2020

A DISSERTATION ON THE TESTING APPROACHES OF AUTONOMOUS CYBER-PHYSICAL SYSTEMS

Cody Behles

A DISSERTATION ON THE TESTING APPROACHES OF AUTONOMOUS CYBER-PHYSICAL SYSTEMS

by

Cody Behles

A Dissertation
Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Liberal Studies

The University of Memphis
May 2020

Acknowledgements:

This research was conducted, in part, with funding by the Air Force Institute of Technology (AFIT) through a subcontract with MacAulay-Brown Inc to the University of Memphis. The research team for the AFIT funded research included University of Memphis faculty Dr. Robin Poston (Principle Investigator), Dr. Mark Gillenson, Dr. Lan Wang, Dr. Eddie Jacobs, and University of Washington faculty Dr. Naveen Chawla. Transcription assistance was provided by Chaitra Anne (PhD candidate, University of Memphis). The Systems Testing Excellence Program (STEP) at the University of Memphis was an instrumental resource in this work.

Abstract:

Testing is among the most critical parts of the development of any system. As technology becomes more advanced, and we continue towards a world defined by integration of advanced features and capabilities (such as being able to think for themselves) in every object, the ability to test these objects becomes infinitely more complicated. This work addresses the testing of autonomous cyber-physical systems (TACPS) by examining current industry best practices as ascertained from interviews with professionals working in the field. Through the interview data provided this work seeks a better understanding of how these systems are tested, the integrated approaches used in testing these systems, and the direction of the industry in the near future. Of interest to readers is investigation of the role of simulation in these testing environments because this work shows how a mix of simulation approaches is combined to overcome development timeline limitations while also addressing a high threshold of safety concerns in a vacuum of clearly defined standards. The results of this work include a series of best practices for professionals performing TACPS and seeks to provide a snapshot of a rapidly evolving landscape defined by emerging technologies that will eventually transform the way all people interact with the physical world. Given the uncertainty and lack of a comprehensive set of defined standards for TACPS, this research seeks to answer the question: How are autonomous cyber-physical systems (ACPS) tested?

Table of Contents

Chapter Page

List of Tables vi

List of Figures vii

Section 1 – Introduction, Background, Defining the Term, and Introducing the Study

1.1 – Introduction 1

1.2 – Defining the IT Process Artifact – Testing of Autonomous of Autonomous Cyber-physical systems 3

1.3 – Introduction to Study 13

1.4 – Literature Review and Current State of the Field 16

1.4.1 - Literature on Testing of Autonomous Cyber-physical Systems 16

1.4.2 Theoretical Approaches in MIS, Methodological Sources, and Posthumanism Approaches 28

Section 2 – Methodology, Study Design and Hypotheses

2.1 – Methodology and Study Design 32

2.2 – Research Questions and Hypotheses 39

Section 3 – Results

3.1 – Introduction 43

3.2 - Code Theme Analysis - Testing 45

3.2.1- Physical Testing, Simulation Testing, and Hybrid System Testing 46

3.2.2 - The Role of Neural Networks in Testing 54

3.2.3 – Value Systems in the Network 56

3.2.4 – Test Cases Discussion 57

3.2.5- Discussions of Testing Culture 59

3.2.6 – Exploratory Testing 61

3.3 – Other Meta-Category Results 62

3.3.1 – System Meta Category 62

3.3.2 – Data Meta Category 65

3.4 - Sentiment Data 67

Section 4 – Analysis and Discussion, Summarized Findings, Challenges, and Best Practices Recommendations

4.1 – Introduction 68

4.2 – Discussion of Hypothesis #1 69

4.2.1 – Safety 70

4.2.2 – Simulation 72

4.2.3 – Test Case Design 76

4.3- Discussion of Hypothesis #2 77

4.4 - Discussion of Hypothesis #3 79

4.5 - Summarized Significant Findings 83

4.6- Recommended Best Practices 85

4.7 - Identified Challenges and Future Research 86

Section 5 - Conclusion 87

Appendices

Appendix A – Survey questions for interviews 99

Appendix B – Code Book 103

List of Tables

Table 1 –Myer’s principles, their interpretation, and application **29-30**

Table 2- Coded Meta Themes by number of references in each interview. **45**

Table 3 – coded reference metadata for “data metacategory” **65**

Table 4 – Sentiment Data Coding Per Interview **67**

Table 5 – Descriptions of Simulation Testing Used in TACPS **72-73**

List of Figures

Figure 1 - Illustration of 5C model with descriptors, **8**

Figure 2 – The Waterfall Testing process outlined. **10**

Figure 3- Deep Artificial Neural Network **21**

Figure 4 – Image testing used to inform the decision-making algorithm **22**

Figure 5 - Word cloud generated from interview data **44**

Figure 6 –Continuum of testing integration **47**

Figure 7 – V Model Testing **52**

Figure 8 - TACPS Simulation Deployment. **74**

Section 1 – Introduction, Background, Defining the Term, and Introducing the Study

1.1 - Introduction

As a dissertation on any topic is by its very nature a heady undertaking rooted deep in the stoicism of tradition and scholarship, it is important to start with the appropriate level of Greek. "The only constant in life is change" said Heraclitus, a thought which has served as the seed for the larger consideration of the role of technology in our world and how this technology impacts and transforms our lives (Kirk, 1951; Schön, 1967). We often talk about how much the world has changed in the last century, but even in the last decade the fundamental shift in our daily experience is dramatic. The pervasiveness of technology has been commented on relentlessly, and I will not repeat that commentary here (Chesbrough, 2017). Instead what I will note, as it is relevant to the larger consideration of any doctorate in liberal studies, is that technology's omnipresent and inescapable role in our world necessitates a deep consideration of the interdisciplinary impact that it brings.[1]

Any individual technology is a convergence of concepts swirled together to create an experience that adds value to our own experiences. To illustrate this simply, we can look at a map. Jerry Brotton discusses in his *A History of the World in 12 Maps* how the "Google map" (a Mercator projection) is imbibed with a mix of cultural bias, technological limitations, and

[1] A great book that explores the interlinkages between the concept of liberal studies and the importance and impact that it has across many disciplines is Fareed Zakaria's "In Defense of a Liberal Education". Zakaria says "Increasingly, thenew core competence is creativity—the right-brain stuff that smart companies are now harnessing to generate top-line growth. It isn't just about math and science anymore. It's about creativity, imagination, and, above all, innovation." (Zakaria, 2015) This is a concept which we find more and more, and reflects fundamentally a concept that is echoed throughout history, must succinctly by Thomas Aquinas who says "Beware the man of a single book". (de Azevedo, 2010)

historical tradition to create a representation of the entire world as we imagine it (Brotton, 2013).[2] This single artifact can be incorporated into other cultural representations (Cui, Yokoi, & Kato, 2009), used in mapping other cultures (Miller, 2006), leveraged to map scholarly impact (Chien, Wang, Chang, & Kan, 2019), enlisted in the archaeology of religious pilgrimage (Meyer, 2019) and a nearly infinite number of other applications, permutations, and considerations. Regardless of the technology being considered, the fact is that maps (or any other technology). will inevitably and inextricably be linked to the universe that created it and that uses it.

We can see the use of this idea (we will call technology-as-artifact) employed most saliently for our discussion in the literature of management information systems. Here, not only is technology-as-artifact given credence in the most esteemed journals, but it is in fact expanded upon so that not only a physical artifact is considered, but processes can also be examined as artifacts (Drechsler, Hevner, & Gill, 2016; Matook & Brown, 2008). The concept of "process" in information systems is core to the foundational literature for the discipline, with the role of the process and the lack of a clear understanding of the information technology (IT) processes within organizations instrumental to early discussions (Ackoff, 1967a; Boland Jr, 1978). Later this idea of the process-as-artifact was institutionalized in the literature as "process improvement," where the focus shifted to the permeability of these artifacts in concert with a larger system, ultimately seeking to create efficiency and profit for businesses (Harrington, 1994). This concept will be central to my discussions later in the work.

[2] Mercator Projection: Created in 1569 by Gerardus Mercator, the Mercator projection is the most common map that we experience in the world today. The map, which was designed for Europeans crossing the Atlantic, distorts the poles so that the more central longitudes are straightened, aiding in said travel. However, this also distorts landmasses making America, Europe, and other western cultural centers proportionally larger than their counterparts at the poles or in the southern hemisphere. Alternative projections, such as the Peters projection, have failed to gain adoption principally because of the biases that favor these centers of power (Haemer, 1949)

First, I need to explain how this work will progress. This work will examine in detail one technology process artifact. The artifact in question is comprised of several constituent parts, I will first dissect the meaning of the artifact in question. I will explore the history and development of each component both in isolation and in conjunction as a single term. I will then move into the core research of this work - the examination of the term through qualitative interviews with practitioners. The interviews that were conducted focused on this process artifact, and the multitude of interpretations and variations that exist in the implementation of the process. From this data I will draw conclusions that can be used to better establish the boundaries and meanings of the process in question. I will endeavor to provide you with guideposts by using traditional components of research – literature review, methodology, conclusion and other expected terms. The goal of this study is to establish best practices and a deeper consideration of the role of the artifact in question.

Section 1.2 – Defining the IT Process Artifact – Testing of Autonomous Cyber-Physical Systems (TACPS)[3]

The process artifact in question for this work is testing of autonomous cyber-physical systems. Within this single artifact are three core concepts that must be expressed at length – autonomy, cyber-physical systems, and systems testing. In addition to individual explorations I will also explore the combinatory effect of using the three separate concepts in a single term.

Artifact #1 – Autonomy and Autonomous Systems

[3] Note that at some points the term autonomous Cyber-physical systems (ACPS) may be used. While TACPS denotes testing of these systems, ACPS refers to the systems themselves.

Automation is fundamentally the concept of self-government - whether in reference to a robot or a republic, the idea that a single unit can create rules, learn from them, modify as needed, and run itself independent of an external actor is universal for autonomous units (Bajracharya, Maimone, & Helmick, 2008). Discussing autonomy in the context of technology inevitably calls to mind the popular culture references and common idols of the concept – HAL 3000, driverless cars, and military drones to name but a few. These images create a false impression that an autonomous future is years or decades away, when in fact the autonomous future has existed in the devices around us for some time. Smartphones are the best synthesis of how this technology impacts everyone - putting the potential for autonomous technology into the hands of everyone. The autonomous Internet of Things (IoT) reality that we live in today is only possible because of the levels of access that we have achieved (Gubbi, Buyya, Marusic, & Palaniswami, 2013). While we may not have autonomous vehicles everywhere yet, their imminence is due to the ubiquity of powerful sensor-based computers in every hand (Krasniqi & Hajrizi, 2016).

The autonomy that I discuss throughout this work stretches back to the Greeks who first used steam powered energy sources to create movement in systems (Smithers, 1997), while others might point to Babbage and Lovelace and their difference engine, the common root of modern autonomy is traced to the 20th century .[4] In 1950 Alan Turing wrote “Computing Machinery and Intelligence,” in which he described for the first time the idea of a digital computer which, when presented with a formal set of rules and the ability to adjust based on

[4] The difference engine(s) was an automatic mechanical calculator that could perform complex polynomial functions using a predefined set of rules. (Swade & Babbage, 2001). It is also the birthplace of “punchcards.” I cannot footnote a footnote, but for those who don’t know punchcards can be roughly described as scantrons or the location of a dimpled chad (Posner, 2000). For a video of the difference engine in action: https://www.youtube.com/watch?v=be1EM3gQkAY

inputs, could respond accordingly. Turing's ultimate goal was for the computer to beat "the imitation game," or be able to effectively mimic a human's actions (Turing, 1950).[5] The point of the game was not to win but instead to establish through thought experiment the goal of a loftier form of autonomy than what was conceived of at the time (Shah, 2011).

Throughout the 20th century humanity has achieved major milestones in autonomy: every single Mars rover has made the journey and conducted research with a degree of autonomy, (Bajracharya et al., 2008; Lim, 1968), we have explored the deepest parts of the oceans (Bandyopadhyay, 2005), we have defeated impossible games (Wang et al., 2016), and have advanced medical science to save lives (Attia, Hossny, Nahavandi, Dalvand, & Asadi, 2018). Regardless of the application, autonomous actions permeate technology. This is crucial to understanding the work in this dissertation. As you will see, the term autonomous system will be used again and again – it is important that you not have a preconceived notion of what that system should look like, or that it should perform a specific function. When I say autonomous system in this paper I can as easily be referring to the controls that would drive a spaceship as I can the decision-making engines that inform a chess robot. It is the presence of autonomy inside the system that matters.

Artifact #2 – Cyber-Physical Systems

[5] There is a great deal of debate around the role of the imitation game and the Turing Test as a useful measure of autonomous function principally because it does not measure intelligent behavior, only how closely that behavior resembles human behavior. One of the core arguments is the measure of human intelligence versus general intelligence. While this does dip into the philosophy of the concept quite a bit, in general we can say that concept of measuring "human intelligence" is flawed for several reasons. One major discrepancy is that some human behavior is unintelligent (lying, making mistakes) and some intelligent behavior is not human (being able to solve complex equations that would reveal the nonhuman status of the machine)(French, 1990; Hayes & Ford, 1995; Hernandez-Orallo, 2000). The ability to hide intelligence which is inherent in the concept as presented (plus other challenges of the Turing Game) can lead to awareness hazards in AI (Yampolskiy, 2016) or the nightmare future of *I, Robot* (Asimov, 2004).

Borrowing from Baheti and Gill, cyber-physical systems (CPS) are "systems with integrated computational and physical capabilities that can interact with humans through many new modalities" (Baheti & Gill, 2011). While this is a succinct definition, the interest of this work is not to investigate fully the breadth of this term as it applies to autonomous systems. Before I narrow the scope of the definition to the specific interests at hand, it is useful for the reader to know what the term can include and how it has impacted the wider development of autonomy and related fields. Most critical for this higher-level summary of the definition above is the core concept contained within Baheti and Gill's definition - human-computer interaction (HCI).

The term "human-computer interaction" gained prominence in the late 1970s and 1980s with the publication of "The Psychology of Human-Computer Interaction". In this first work, the authors consider what is perhaps the greatest challenge of HCI – the conflict between technical limitations, user capability, system requirements, and human understanding (Card, 2018). The author's state simply that HCI "is easy to find in a gross way – just follow a data path outward until you stumble across a human being." The term as initially applied existed in a world of terminal displays and rudimentary programming engagements for computers. As HCI grew more complex so did the places in the system where one might have stumbled upon human beings. This evolution, as explained by Jonathan Grudin, occurred both in the machines but also in the people who used them. The expectations and experiences of the users in their interface requirements pushed the envelope ever further forward (Grudin, 2008).

Autonomous system HCI may seem oxymoronic, but it is in fact essential to operation. The literature on the study of autonomous HCI, emphasizes that beyond the developmental period it is critical to sustained operation and mission success that HCI be an integral operational component (Stumpf, Burnett, Pipek, & Wong, 2012). This manifests itself most often in maintenance and in the testing of autonomous cyber-physical systems (ACPS) (Durst & Gray, 2014; Thompson, 2008). I will discuss testing conceptually more in the next section, but now I will spend a moment discussing how cyber-physical systems are deployed.

Among the most common models for autonomous system deployment is the 5C (configure, cognition, cyber, conversion, connection) architecture. The 5C architecture for deploying cyber-physical systems in manufacturing applications is a widely utilized framework for integrating cyber-physical systems because it establishes a common language for discourse (Lee, J., Bagheri, & Kao, 2015). 5C is also used outside of manufacturing and so serves as a useful framework for the TACPS discussion (Xu, Xu, & Li, 2018); the standards and classifications for integration of all autonomous and cyber-physical systems begins partly through this model, and the 5C model also helps creates an inter-industry dialogue.

Briefly summarized the 5C model is a five-step process (see Figure 1):

- Step one is focused on being able to acquire reliable data from the machines and includes the ability to connect consistently and reliably;
- Step two is focused on inferring meaningful information from the data;
- Step three, the cyber level, is the central information hub of the model, and would be where the principle machine learning analysis occurs;
- Step four, cognition, presents the information acquired back to the user and
- Step five (configuration) acts as a resilience control system (RCS) and applies corrective supervisory control to the system. The model was further modified in a

second work to operate without user interface for self-aware machines.(Bagheri, Yang, Kao, & Lee, 2015)

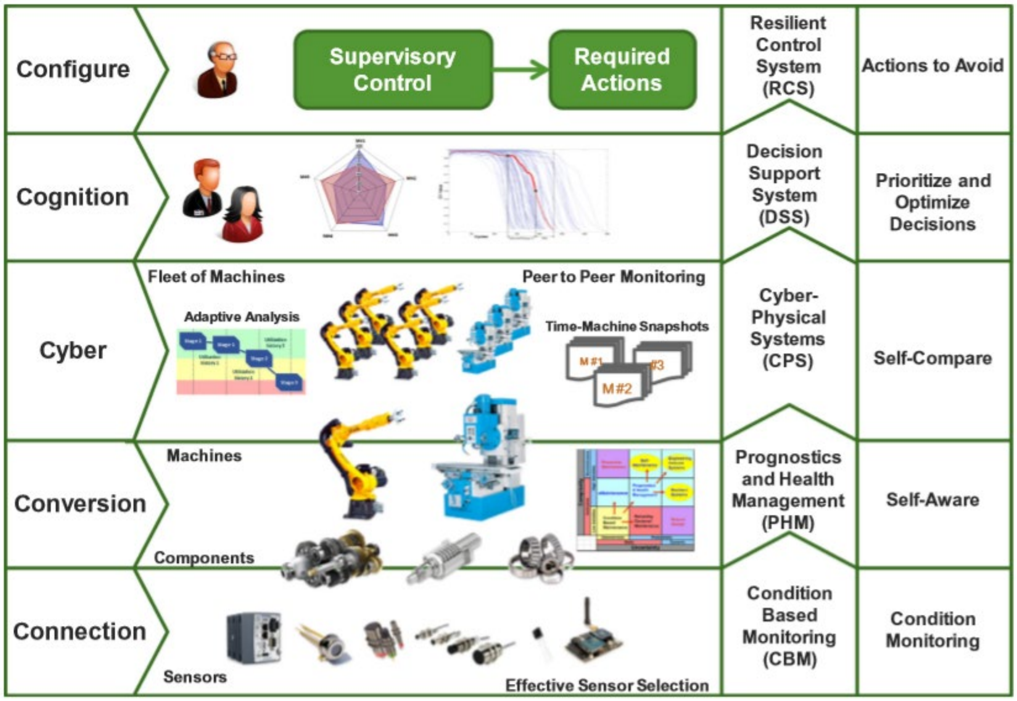


Figure 1 - Illustration of 5C model with descriptors, from Baghari and Kao, p.20 (Bagheri et al., 2015)

Understanding the 5C in the context of defining cyber-physical systems is important. Not only do I want to ensure that you as the reader understand what systems are being discussed, but also how they are deployed in real world scenarios. In this study I will not limit the definition of autonomous cyber-physical systems to specific tasks that are carried out by the machine. While I initially considered at length specific applications of mobile systems including last mile delivery (Boysen, Schwerdfeger, & Weidinger, 2018), pick and place robots (Zeng et al., 2018) autonomous and semi-autonomous fleet management (Petty, 2017; Switkes, Gerdes, &

Berdichevsky, 2019), autonomous drone delivery systems (Brunner, Szebedy, Tanner, & Wattenhofer, 2019), and other types of vehicular autonomous cyber-physical systems, in the end it did not make sense to limit the discussion. There was also the practical concern of finding participants if a more limited set of CPS were the subject of this research.

Artifact #3 –Testing

Systems testing is the testing conducted on a complete integrated system to evaluate the system's compliance to specific requirements. [6] This is to be distinguished from analog testing, beta testing, or other terms that are often either focused on physical performance or developmental stages (Lucas, Ginzberg, & Schultz, 1991). The concept of testing is foundational to the management of large systems. Ackoff, whose work I mentioned earlier, laments that in their pursuit of building systems that work effectively for managers, system designers often obfuscate important details to hide their complexity (Ackoff, 1967). Kast and Rosenzweig outlined early on the need for an iterative approach to building systems and highlight the early need for input from multiple sources (Kast & Rosenzweig, 1972). Other authors too, discussed the importance of iteration and inputs in order to build the best systems (Drucker, 1988; Galbraith, 1974; Nolan, 1979). It is important here that all these early sources Ire engineering focused. Later in this work I will further explore the relationship between engineering and testing.

[6] For those without any background in systems testing, it is important to distinguish between testing generally and systems testing. An analogy that might be helpful is to consider the manufacturing of a physical object like a pen. During the process of manufacturing a ballpoint pen, the cap, the body, the tail, the ink cartridge and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed. When the complete pen is integrated, System Testing is performed.

Systems testing and software testing are different things, but because a software that controls a system inherently includes all of the functionality that the software is intended to control, you will see the terms used in the same work frequently.

For cyber-physical systems, the road to the current state of testing is rooted in the trajectory of testing development for management information systems. Testing in the information technology context was originally designed to help improve the efficiency of management information systems. The iterative nature of this process led to the adoption of the first widely used model for software development - waterfall. First introduced by Herbert Bennington in 1956 and later modified and widely consumed using Winston Royce's 1970s model (Royce, 1987), the Waterfall model helped to address many of the concerns raised in early Management Information Systems (MIS) works that pointed out the need for feedback in the product development lifecycle. The waterfall method remained the dominant model for systems development until the mid-2000s. Included below is an image from Royce's paper which outlines the steps for a waterfall project lifecycle.

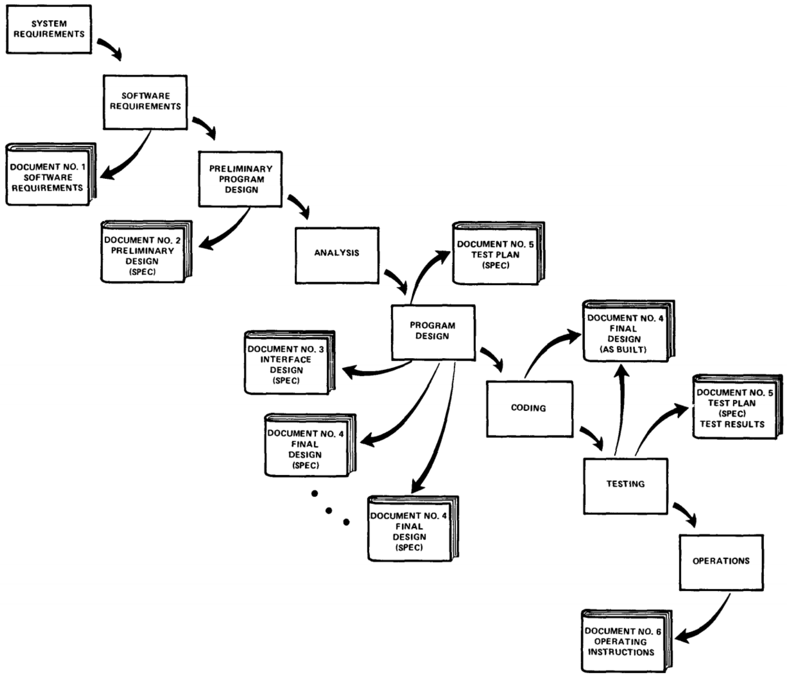

SYSTEM REQUIREMENTS
SOFTWARE REQUIREMENTS
DOCUMENT NO. 1 SOFTWARE REQUIREMENTS
PRELIMINARY PROGRAM DESIGN
DOCUMENT NO. 2 PRELIMINARY DESIGN (SPEC)
ANALYSIS
DOCUMENT NO. 5 TEST PLAN (SPEC)
PROGRAM DESIGN
DOCUMENT NO. 3 INTERFACE DESIGN (SPEC)
DOCUMENT NO. 4 FINAL DESIGN (AS BUILT)
CODING
DOCUMENT NO. 5 TEST PLAN (SPEC) TEST RESULTS
DOCUMENT NO. 4 FINAL DESIGN (SPEC)
TESTING
DOCUMENT NO. 4 FINAL DESIGN (SPEC)
OPERATIONS
DOCUMENT NO. 6 OPERATING INSTRUCTIONS

Figure 2 – The Waterfall Testing process outlined. From Royce, page 30 (Royce, 1987)

The challenges that those early management information systems (MIS) scholars grappled with are specifically addressed in the waterfall model. At the beginning of the lifecycle is the development of requirements. This is, for example, a manager in a company saying they need report x generated every day. The requirements are then digested through preliminary program, then analysis, then the design of the solution itself, followed by the actual coding of the program, then testing it to make sure it works, then putting it into production. This approach is not only the backbone of many major software development, but also shares common features with the 5C model for manufacturing. The iterative and improvable nature of manufacturing system integration shares a common thread with the software system development process. Waterfall does have limitations, and as software development efforts become bigger and more projects need to be moved through the development lifecycle, the waterfall model begins to fall apart.

Throughout the late 80s and into the 90s, the waterfall methodology begins to give way to what would eventually become an Agile methodology (Gilb & Finzi, 1988). Several works contribute to the evolution from waterfall to Agile in popularity but a clear step towards wider adoption of the Agile model comes from Kaner, Faulk and Ngyuen. Recognizing that software testers cannot change a corporate development philosophy, they propose an evolutionary approach to testing which principally focuses on scheduling learning about the problem *before* thinking about the problem and responsiveness to change in specifications and requirements (Kaner, Falk, & Nguyen, 2000).

The emphasis on restructuring the thought process to focus on development would go on to serve as a foundational element of the Agile movement. Agile methodology can be summarized as retooling of the development process to be less linear, instead emphasizing the continuous development approach and encouraging a concurrent strategy to development (Crispin & Gregory, 2009). The Agile methodology is an important shift in the software development process but while it added a great deal of value to the earlier steps in the development lifecycle, testing was still largely relegated testing to the end of the cycle. This was brought into stark relief in 2011 when Onita and Dhaliwal identified the misalignment between developers and testers, recommending that by integrating the testers earlier into the development lifecycle, some of the more costly challenges associated with IT development (such as missed deadlines) could be mitigated (Dhaliwal, Onita, Poston, & Zhang, 2011). Today developers are motivated to integrate testing at all stages of the development process. The continuous improvement/continuous delivery (CI/CD) movement nearly mandates an integrated approach to testing (Lewis, 2017). Within the manufacturing space, the trajectory of testing has mirrored in many respects the trajectory of testing in software development. Functional testing (FCT) is typically conducted in the last phase of product line development and this areas while scan testing, LBIST (logic built in self-test), on-chip testing, cache memory testing, and i/o testing are all integrated in various stages of the production lifecycle (Kundu, Mak, & Galivanche, 2004). In practice the testing methodologies of software systems are being married with the scaled deployment of autonomous systems, but there has been little research exploring the specific nature of the approaches taken by the adopters.

Convergence of the Artifacts

It is important that the reader understand the constituent components of the term “testing of autonomous cyber-physical systems” (TACPS). It is only in fully understanding the term that this work can begin to consider what this study is grappling with. Not only are three different concepts being considered simultaneously, but each concept is rooted within a different scholarly research trajectory. Earlier I mentioned the power of placing a sensor-based device in every hand. While this was a breakthrough for advancing autonomous capabilities, it was also a breakthrough for the testing of autonomous cyber-physical systems. When the smartphone changed the concept of user interface, it also changed the concept of a cyber-physical system. When the smartphone changed how quickly we receive information and how rapidly companies can improve devices and services, it also changed systems testing (Canfora et al., 2013). This term cannot be divorced from the culture in which it resides, and it would not be possible without the innovations of the last two decades.

Section 1.3 – Introduction to Study

With background on the terms now established, I will push into the core of what this work will consider. As mentioned above, there are three distinct traditions in the term TACPS. The autonomous cyber-physical system itself is a convergence of different traditions – engineering, software and management information systems, HCI, and much more. The complexity of these devices means that many different people with many different perspectives and training in many different disciplines are integral to their development. The exploration of this concept has been written about most notably by Michael Decker in his study of technology assessment methods (Decker & Grunwald, 2001; Decker, 2008), while others have focused on the platform nature of these technologies as an interdisciplinary function (Boix Mansilla, Lamont, & Sato, 2016; Valdez, Schaar, Ziefle, & Holzinger, 2014).

The disciplinarily heterogeneous constitution of autonomous systems means that when it comes to creating synthesized approaches, best practices, and other industry normative techniques, there is no single adopted strategy. Anytime that we are discussing a technology at the cutting edge, it is important to consider the role of innovation within large organizations.

Traditional innovation ecosystems in these types of environments are not efficient at distilling the ideas that could lead to industry standards which means that sometimes best practices exist but are never adopted among wider groups or organizations (de Vasconcelos Gomes, Leonardo Augusto, Facin, Salerno, & Ikenami, 2018). Despite this, the integration of ideas from outside firms has been shown to be a critical requirement of long-term success and innovation growth for companies. [7] (Trantopoulos, von Krogh, Wallin, & Woerter, 2017). In order for best practices in autonomous cyber-physical systems (or any other innovative technology or process) to be adopted and shared widely they must be investigated further. This line of thinking is especially salient, as will be discussed later in this work, when we consider that there are no safety standards in place which specifically address specifically ACPS. Both engineering and MIS have been shown to leverage absorptive innovation strategies in the adoption of new ideas (Hameed, Counsell, & Swift, 2012), and so for TACPS (which draws on both engineering and MIS traditions) the likelihood that best practices and innovative strategies intermingle testing best practices from both traditions is high. It is important to remember that these disciplines (engineering and systems testing) share a common root, as outlined above in the discussion of history of the term systems testing. Based on the interview results, in TACPS I see the two disciplines recombine in interesting ways. Testing is a natural way to examine the intersection of these disciplines, because it is an essential part of both and it is always at the forefront of challenges that developers face, because it is through testing that those challenges are discovered. Yet, because they have different approaches, they also perceive challenges differently.

[7] For those without MIS or business discipline background – "A firm is a for-profit business organization—such as a corporation, limited liability company (LLC), or partnership—that provides professional services. Most firms have just one location. However, a business firm consists of one or more physical establishments, in which all fall under the same ownership and use the same," Will Kenton, *Investopedia, Accessed April 2020*

In this work I will examine the testing approaches of autonomous cyber-physical systems. The work will employ qualitative methodologies with a consideration of posthumanist approaches and an emphasis on the role of technology and inanimate features in the analysis of the data.[8] Through the analysis of interviews joined with related research and information on systems testing, this work will generate a guideline for best practices around the testing of autonomous cyber-physical systems, synthesized from interviews with industry practitioners.

Study Design

The study will utilize a qualitative semi-structured interview approach to collecting data. There are several approaches to questioning in semi-structured interviews. For this study, I will adopt the ethnographic approach to questioning outlined in Leech's 2002 summary of question types that allows for exploration without preconceived knowledge of how testing is conducted in the facilities (Leech, 2002). Specifically, I will use what Spradley called *grand tour questions,* and will structure the interview around providing the interview opportunity to provide at length responses structured around the way they build their questions (Spradley, 1979).

The interviews will be conducted with industry professionals who are experienced with the integration of autonomous systems. Due to the nature of trade secrets associated with some of these implementations, the interviews were anonymized and the data treated and digested in aggregate. Transcripts of the interviews are not attached but may be read upon request if required for the purposes of the dissertation review. The collection of the interviews was done with a second researcher in order to corroborate results. Using the conventional inductive qualitative

[8] I will spend more time in the literature review on explaining in more detail the post-humanist qualitative approach, but in brief here I will note that there are a host of different interpretations of post-humanist methodology. In this context I intend to limit my focus to those aspects of posthumanism theory which decenter the human experience in favor of the system and technology as equal components in the vein of scholars like Bruno Latour (Latour, 2000).

analysis methods, the results of the interviews were coded, and from that code a theoretical framework and best practices for TACPS will be developed and suggested.

In order to effectively communicate the results so that they are digestible to the largest audiences, this work will adopt some of the methodologies and techniques from the scientific communication literature (Weigold, 2001). As you will see in the history of testing below, there is a precedent for the reconsideration of the role of testing in existing development models, and the opportunity to reconsider TACPS.

Section 1.4 – Literature Review and Current State of the Field

While I have summarized some background details on the components of the term TACPS, more critical to this work is understanding the background of testing of autonomous cyber-physical systems. Certainly this is not the first work to consider this concept, but given that relatively little research that has been done to analyze the current state of testing autonomous cyber-physical systems in industry, it does fill a considerable gap in the field.

Section 1.4.1 - Literature on Testing of Autonomous Cyber-physical Systems

I covered previously the evolution of testing, and I will not repeat here. Instead I will spend more time considering the testing protocols and techniques currently employed. It is helpful for the reader to know the types of testing employed in the development and maintenance of autonomous cyber-physical systems. To summarize I have included here brief definitions to level set the terminology:

Test-first Development (TFD): Also known as test driven development (TDD) is based on formalizing a piece of functionality as a test, implementing the functionality such that the test passes, and iterating the process. The test is often written before the program itself (Beck, 2003).

Black Box Testing: also known as functional testing, is testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and conditions (Gao, Tsao, & Wu, 2003).

Equivalence Class Testing: is used by the team of testers for grouping and partitioning of the test input data, which is then used for the purpose of testing the software product into a number of different classes. These different classes resemble the specified requirements and common behavior or attribute(s) of the aggregated inputs (Burnstein, 2006).

Boundary Value Testing: Similar to equivalence testing in that data are partitioned, but the partitions include representatives of boundary values in a range. These boundaries are the common location of errors that result in faults in the program and so this testing zeros in on these potential faults.

Pairwise Analysis Testing: Requires that for each pair of input parameters in a system, every combination of valid values of these two parameters be covered by at least one test case (Tai & Lei, 2002).

White Box Testing: It is the detailed investigation of internal logic and structure of the code. In white box testing it is necessary for a tester to have full knowledge of source code which is the opposite of the black box, where the tester has no knowledge (Nidhra & Dondeti, 2012).

Grey Box Testing: White box + Black box = Grey box, it is a technique to test the application with limited knowledge of the internal working of an application and has the knowledge of fundamental aspects of the system. (Nidhra & Dondeti, 2012).

Branch Coverage: a requirement that, for each branch in the program (e.g., if statements, loops), each branch have been executed at least once during testing.

Exploratory Testing: Simultaneous learning, test design, and execution. In other words, exploratory testing is any testing to the extent that the tester actively controls the design of the tests as those tests are performed and uses information gained while testing to design new and better tests (Bach, 2003).

The terms above are meant to serve as a condensed summary of the expansive landscape of system testing. They also all serve as extensions of what is the basis of software testing and evaluation - functionality testing. This method of testing assesses whether the software correctly performs the operations it was designed for (Hass, 2014). You can see features and versions of this laced throughout the definition**s** above. In addition to functionality testing and its elements, software testing and evaluation includes performance testing, security testing, usability testing, and internationalization testing.[9]

Regardless of whether they move or not, the mutable nature of an autonomous cyber-physical system means that the program the tester sees today could be different than the one they test tomorrow (Aniculaesei, Grieser, Rausch, Rehfeldt, & Warnecke, 2018). The origins of

[9] These four testing types are not included in this study. Though each is integral to the development of a completed system, they generally require domain specific knowledge which was beyond the scope of the questions put forward during the interviews. Please see appendix A for an example of the survey questions.

software testing and evaluation by contrast focused on “static” systems in the sense that a set of input variable values produces a reliable, correct output. The comparative dynamism of autonomous systems require a rapid succession of decisions and outputs based on a continuous stream of sensor input data; the number of possible “test cases” is virtually infinite (Ryan, 1999). Since the field is so new, there is relatively little sound literature so I will forgo breadth for an emphasis on depth - examining a handful of seminal works in detail.

DeepTest, DeepXplore, and DeepGauge

There are several other key differences between autonomous cyber-physical systems (CPS) and traditional software testing approaches. The first is the presence of machine learning and artificial intelligence within the systems. An important trio of papers that grapples with this is the *DeepTest, DeepXplore*, and *DeepGauge* works that examines the deep neural network implementations of autonomous vehicle systems and explores solutions for testing these systems in light of the fact that “it is hard to build robust safety-critical systems only using manual test cases” (Ma et al., 2018; Pei, Cao, Yang, & Jana, 2017; Tian, Pei, Jana, & Ray, 2018a). For most of this section I will focus on *DeepTest*, with a brief overview of the other two works at the end. Fundamentally all three works share common premises that are shared throughout.

Tian, et al. focus on an aspect of the software in autonomous CPS that represents a relatively new use of artificial neural networks (ANN) (Norvig & Intelligence, 2002). This new phenomenon is the embedding of an ANN in application software, in this case specifically in autonomous CPS control software. The use of ANNs provides a machine learning capability that allows an autonomous CPS to gradually improve its navigation and object avoidance in its environment. The authors of the *DeepTest* paper speak of deep neural networks (DNN), meaning

ANNs with multiple neuron layers.[10] They speak of progressive neuron layers that start with small features of the environment, then build to complete objects. Some variants allow loops between layers (Figure 3).

ANNs must be trained with sets of inputs with expected outputs. As with any process, there are a variety of recognized procedures and cautions in training ANNs. Tian, et al., speak of training the ANNs with "synthetic images." These are sets of data intended to simulate real world environments and obstacles. They begin with "seed images," then apply transformations to the seed images to produce new images, which is a form of mutation (Figure 4). For example, they add or subtract a constant to each pixel of an image to produce a new image. The expected output (i.e. what the autonomous CPS should do given a set of inputs) of each seed image has to be established. In many cases, the expected output of transformed (mutated) images is also known. For example, if the transformation involves different lighting conditions, the autonomous CPS should perform in the same way for each such variation. However, there can be acceptable variations in the expected behavior. For example, slight variations in the steering angle may be acceptable.

[10] The terms "DNN" and "ANN" will be used interchangeably throughout this work.

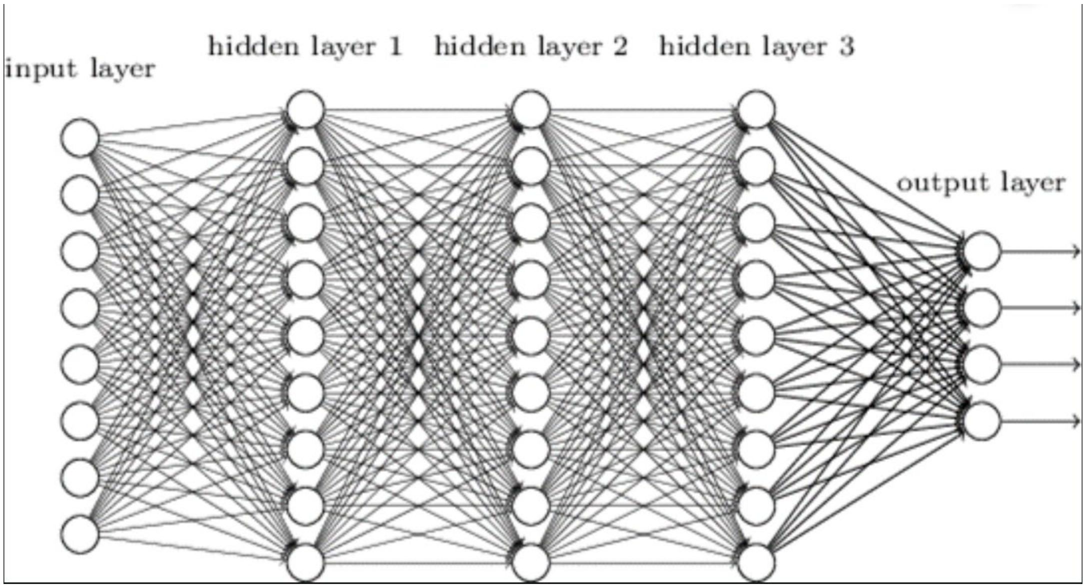


Figure 3- Illustration of the Deep Artificial Neural Network used in the DeepTest work (Tian, Pei, Jana, & Ray, 2018)

In traditional white box software testing, there are the concepts of statement coverage, branch coverage, and path coverage. These refer to the number of statement, branches, and paths, respectively, that a set of test cases traverse. Tian, et al., leverage these coverage concepts to establish "neuron coverage" as a means of determining how many neurons in the ANN a set of test images traverses. They say that neuron coverage is, "the number of neurons activated by a set of test inputs." They go on to say, "Changes in neuron coverage are statistically correlated with changes in the actions of self-driving cars." Since exhaustive testing is impossible in these complex systems, neuron coverage is used to partition the I/O space into equivalence classes based on the assumption that inputs with similar neuron coverage are part of the same equivalence class. That is, neuron coverage is correlated with different actions that an autonomous ANN can take. They also determined that their process of starting the ANN training with seed images and then applying transformations to them to produce additional images increases neuron coverage.

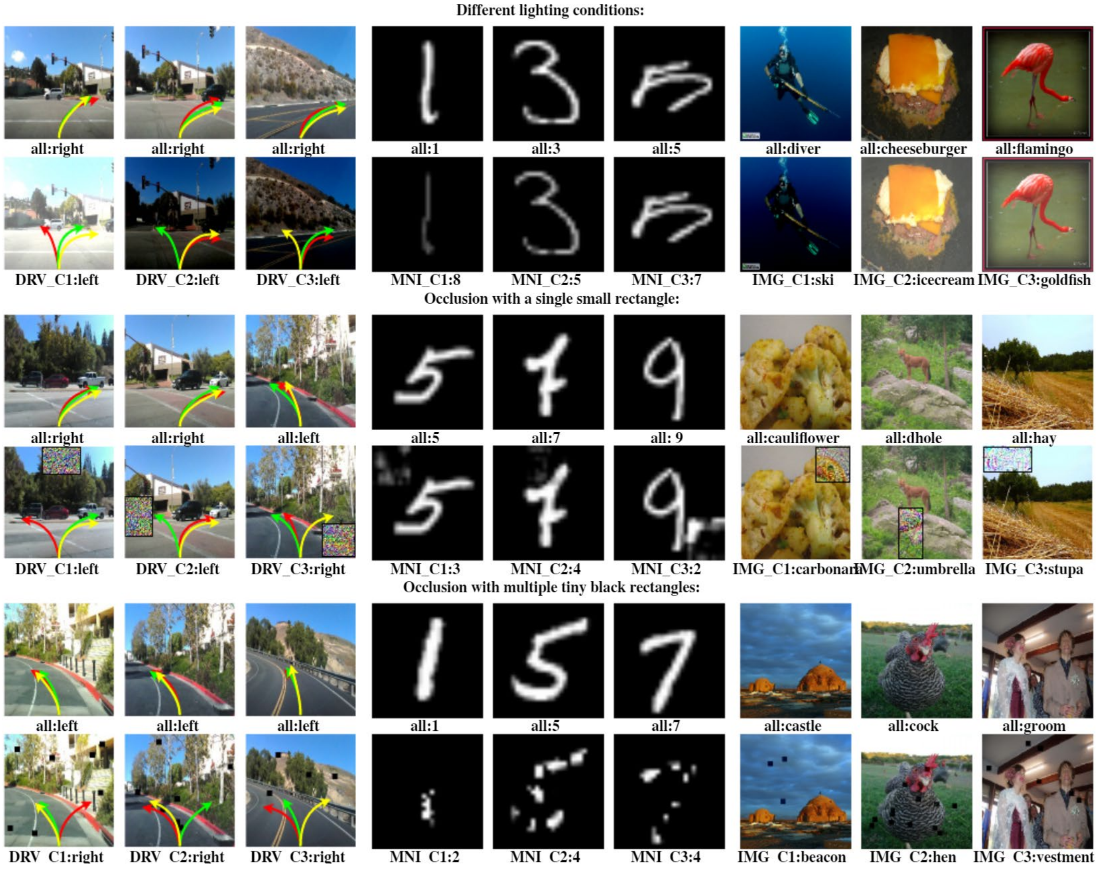


Figure 4 – Image from DeepXplore paper showing the image testing used to inform the decision-making algorithm and degrees of variance (Pei, Cao, Yang, & Jana, 2017)

DeepXplore, one of the two companion articles to DeepTest, is the white box framework for systematically testing Deep Neural Networks (DNN) for erroneous corner case behavior. An important takeaway from the work was that the researchers found that neuron coverage is a significantly better metric than code coverage for measuring DNN outputs. The work also eradicates the need for training DNN, performing in milliseconds what an image trainer would take a week to do (Pei et al., 2017). DeepGauge is the third work and focuses on multigranularity testing criteria for deep learning systems aiming to render a multifaceted portrayal of the test bed.[11] The DeepGauge fills in where the DeepXplore lacked in differentiating between original test data and adversarial test data (Ma et al., 2018).

Challenges in Autonomous Vehicle Testing and Validation

In their 2016 work, Philip Koopman and Michael Wagner explore other challenges around TACPS: driver "out of the loop," complexity of requirements, fault injection and systemic non-determinism (Koopman & Wagner, 2016). First, with the driver "out of the loop," the system must take on all exception-handling responsibilities. This requires a high Automotive Safety Integrity Level (ASIL), as described in ISO 26262[12]. They go on the point out that the requirements for the software in the ACPS to handle are too numerous and complex to possibly list in advance. Consider, for example, all the possible obstacles, lighting conditions, and weather conditions that one encounters on the road. Add to that different types of roads, different off-road conditions, and different terrain. Then, consider the possibility of encountering multiple objects at different angles from the vehicle and at different speeds in different road, weather, and lighting conditions, and the number of combinations of these clearly support Koopman and Wagner's statement that the requirements are too numerous and complex to possibly list in advance.

The authors also note that a "fail-silent" (also known as "fail-stop" or "fail-safe") response to the detection of a serious error may not be feasible. Instead, a "fail-operational" response may be required. Consider an autonomous moving vehicle that suddenly discovers an

[11] For those without testing background, a test bed is a typically virtual simulated environment that mimics real world conditions or other parameters of a system. In this case, a road environment for a vehicular system.
[12] The safety standards here (ISO 26262) is "road vehicles-functional safety) and is adapted from IEC 61508 (functional safety for automotive systems. The Frankenstein's monster that is this standard is indicative of a lack of uniform and originally designed safety standards for autonomous CPS

object ahead that it must avoid. It is not enough to avoid the object. If it is traveling at 60mph, it must avoid the object and at the same time determine how it will continue traveling on its designated path, which may even involve avoiding other objects. It doesn't have the luxury of avoiding the first object and being "done." It is in continuous operation. This also brings up the issue of artificial intelligence embedded in the software exercising ethical decision making. If it must make a choice between two paths, each of which may cause people on the road or in the vehicle harm, which one does it choose? This is referred to as the well-known "trolley problem."

An intriguing solution to the problems described above is to employ low ASIL actuators with a high ASIL monitor. That is, concede that not all situations can be anticipated in the requirements and in the software testing, and employ another layer of software that monitors the vehicle's actions and stops it from doing something dangerous. Unfortunately, this brings us back to the fail-silent versus fail-operational conundrum as the monitoring software would have to be integrated with the rest of the software in such a way that it would have to return control to the software operating the actuators as soon as it prevented the dangerous action. This may not be easily accomplished. It may require redundant systems that do not have redundant inputs or that respond to redundant inputs in the same way.

Interestingly, this concept of allowing software to "get into trouble" and then having other monitoring software catch it and stop it is not without precedent in information technology. A classic problem in information technology occurs when two transactions attempt to update one or more pieces of data at the same time, which can result in inaccurate data (Hoffer, Ramesh, & Topi, 2019). A process known as "concurrency control" was developed many years ago that employs "locks" to ensure that one transaction completes its updates while preventing other transactions from updating the data it is updating at the same time. Unfortunately, this “pessimistic” approach can be inefficient from a performance standpoint. So, a more “optimistic” approach, known as “versioning,” was developed (Nystrom, Nolin, Tesanovic, Norstrom, & Hansson, 2004). Versioning assumes that most of the time there will be no such conflicts between transactions. Each transaction makes updates to the data, not in the database itself, but in its own holding area. These updates are timestamped. Monitoring software (the functional equivalent of the monitoring software described above for the autonomous CPS) then allows the database to be updated if there is not conflict between transactions. If there is a conflict, i.e. two transactions are trying to update the same data at the same time, the one with the earlier timestamp is allowed to update the database while the other is forced to start all over again with the newly updated data (Chatterjee, Arun, Agarwal, Speckhard, & Vasudevan, 2004).

Another approach to testing and evaluating the software in the autonomous CPS that is suggested by Koopman and Wagner is a phased expansion of requirements. For example, begin utilizing highways in daylight and clear weather. Then, gradually vary the expectations of the software by introducing other terrain, weather, and lighting environments, new obstacles, and new goals. Clearly, this gradual approach to increasing the complexity that the autonomous CPS has to deal with, also means that testing its software can follow a gradual approach and will be much more effective. Again, there is precedent for this in information technology. Earlier in this work I spoke of the Agile approach to software development/acquisition and testing and evaluation. One of the precepts of Agile software development is that development and testing begins with the “critical path” through the application, i.e. the application’s central core requirement. Then, features are gradually added to the software and tested on a priority and risk basis (Gillenson, Racer, Zhang, Booth, & Dugan, 2016).

Yet another well-known approach to software testing that Koopman and Wagner suggest is “fault injection.” This involves inserting exceptional conditions as a way of further validating a system that has passed traditional functional testing. Fault injection can be performed to test the software of any application, be it a static application or a dynamic application such as the control software for an autonomous CPS. Indeed, I speak of testing and evaluating “edge conditions” as a way of validating the software in autonomous CPS and this is a form of fault injection. This also comes under the heading of “exploratory testing” (Hass, 2014)

Robustness Testing of Autonomy Software

The third work included here focuses more on the internals of the software and how messages and data move across internal communications channels, directed by multiple controllers (Hutchison et al., 2018). They discuss “stateful” operation in which internal states are created and updated as the autonomous CPS follows a plan and moves through its environment. They also describe “temporal” operation involving sequential requirements with data gathered to direct the autonomous CPS to the next point in its planned path.

Some of Hutchison, et al.’s comments are quite similar to Koopman and Wagner’s. For example, they talk about “safety-critical” systems with control loops compensating for errors. So, interestingly, while Koopman and Wagner describe a “multi-layered approach” with specialized monitoring software stopping the autonomous CPS from doing something dangerous, Hutchison, et al., describe a “control loops approach” to accomplish the same goal. The concept of fault injection as a software testing method is common to both papers. Hutchison, et al., describe it as injecting exceptional values at component interfaces. With their focus on message communication in the autonomous CPS on internal communication channels, they also describe “interception testing” in which values of the messages being sent over the internal network are manipulated or mutated. Mutating values in test cases, in this case in the form of internal messages, is a classic form of exploratory software testing (Hass, 2014).

A Collection of Software Engineering Challenges in Big Data Systems

A fourth paper that should be considered in this deep dive on autonomous systems testing literature is the recent work outlining challenges in engineering these systems proposed by Oliver Hummel (Hummel, Eichelberger, Giloj, Werle, & Schmid, 2018). While the work is not explicitly addressing testing in the way the other works highlighted do, the challenges identified have ramifications for testing in real ways. Principally they are divided into four major categories – project and requirements management, architecture and development, quality assurance, and deployment and operations. All of these categories have salience for testing but importantly bring an engineering perspective. As mentioned at the beginning of the work, the intersection of engineering and MIS is a crucial component of this analysis.

On project and requirements management, Hummel discusses the interesting challenge of grappling with the magnitude of these systems as a key obstacle. The complexity can lead to unclear requirements, as the total functionality of the system is not always apparent. Several of the emphasized challenge areas focus on the fact that coordinating diverse skillsets and tradeoffs between quality and performance is a challenge. The author emphasizes the importance of testing in reducing the risk of unforeseen project management challenges. In quality assurance, where testing is often employed for critical feedback on the correctness of outputs. The author also addresses the hardware limitations for testing. “…thorough testing of Big Data systems requires a similar workload as will be used in the real system, and testing also issues such as parallelization, performance, scalability, etc. For this purpose, an appropriate test system comparable to the production system should be available" (Hummel et al., 2018).

Section 1.4.2 - Theoretical Approaches in MIS, Methodological Sources, and Further Information on Posthumanism Approaches

As mentioned previously, for this study I will employ qualitative analytical methodologies in the approach. While qualitative approaches are not the preeminent methodological approach (Lee, A. S., 1989), they have become a common feature in top tier journals, especially when employed in a mixed methods approach (Kaplan & Duchon, 1988; Palvia, Kakhki, Ghoshal, Uppala, & Wang, 2015; Trauth, 2001). In this section I will explore in greater detail the methodological tradition that informs the interview process that was conducted, and highlight the use of posthumanism in this context.

Among the more impactful works examining the inclusion of qualitative methodological approaches in MIS is the work of Michael Myers. Among his notable works on the topic is "A set of principles for conducting and evaluating interpretive field studies in information systems" which outlines in seven principles the execution of field studies in MIS (Myers, 1997b). Outlined in the table below in brief, Myers says these principles help to fix measurable quality into interpretive research by deriving the principles from anthropology, phenomenology, and hermeneutics.[13] For the purposes of this study I have incorporated into the chart the specific interpretation and application of the principles for this work (table 1).

[13] For those without deep qualitative literary exposure:
Phenomenology: the study of structures of consciousness as experienced from the first-person point of view. The central structure of an experience is its intentionality, its being directed toward something, as it is an experience of or about some object.

Table 1 – An outline of Myer's principles, their interpretation (by Myers), and my application of each principle in this study (Myers, 1997b)

Myer's Principles	**Interpretation**	**Application in this study**
The Fundamental Principle of the Hermeneutic Circle	Suggestion that all human understanding is achieved by iterating between considering the interdependent meaning of parts and of the whole that they form. Fundamental to all other principles	Each interview with a systems tester contributes to a portion of the narrative that I construct as recommendations for best practices
The Principle of contextualization	Requires critical reflection of the social and historical background of the research setting, so that the intended audience can see how the current situation under investigation emerged	Each interview should consider the background of the tester. In this case, I am limiting this information to their basic biographical information and the fact that each company has their own applications of TACPS
The Principle of Interaction between the Researchers and Subjects	Requires reflection on how the research materials are socially constructed through the interaction between the researchers and participants	Consideration of the way the questions Ire selected, including possible biases or preconceptions of the creators that would preclude certain details from being inquired upon
The Principle of Abstraction and Generalization	Requires relating the ideographic[14] details revealed by the data interpretation through the application of principles one and two to theoretical, general concepts that describe the nature of	Identifying the limitations in the interviewee's knowledge based on their own interpretation of the questions at hand. Interpreting the knowledge shared into a model or recommendations

[14] Idiographic: based on what Kant described as a tendency to specify, and is typical for the humanities. It describes the effort to understand the meaning of contingent, unique, and often cultural or subjective phenomena.

Table 1 Continued

	human understanding and social action	
The Principle of Dialogical reasoning	Requires sensitivity to the possible contradictions between theoretical preconceptions guiding the research design and the actual findings with subsequent cycles of revision.	The scholarly articles may not align with the actual experiences of those conducting the tests, or experiences may yield new sources of information for further investigation
The Principle of Multiple Interpretations	Requires sensitivity to possible differences in interpretations among participants as are typically expressed in multiple narratives or stories of the same sequence of events under study.	The common experience of testers working in similar environments may differ, and in that difference, I can find information to inform the model.
The Principle of Suspicion	Requires sensitivity to possible "biases" and systematic "distortions" in the narratives collected from the participants.	Understanding that background and other factors could limit certain areas of knowledge around testing that should be considered.

When conducting interviews on technology subjects, the posthumanist approach is commonly applied. Posthumanism is often a misunderstood term and frequently smashes together divergent concepts that can include a myriad of approaches and concepts that may or may not be within the posthumanist framework. For this work I will adopt a few key interpretations in the methodological application of the concept. The first is among the foundational texts in the field, the work of Bruno Latour. Specifically, we will highlight the approach in his work where Latour speaks at length about the role of humans and non-humans in the interview process. He says,

> “To speak of ‘humans’ and ‘non-humans’ allows only a rough approximation that still borrows from modern philosophy the stupefying idea that there exists humans and non-humans, whereas there are only trajectories and dispatches, paths and trails. But we know that the elements, whatever they may be, are substituted and transformed” (Latour, 2000)

Here the author is arguing that it is a fallacy to consider that in the context of an interview that we can draw a distinction between the human and the machine – that they must exist together in the context of the interview and the posthumanist methodological approach. We cannot interview a person without equally considering the devices and technologies that they use, the culture and experiences of the interviewer, and the culture and experiences which informed the creation of the machine or system we are observing. Through a posthumanist interview process, the artifact is given agency(Latour, 2000).

Building on the ideas proposed by Latour, Cary Wolfe in the work *What is Posthumanism* contributes,

> “…when we talk about posthumanism, we are not just talking about a thematic of the decentering of the human in relation to either evolutionary, ecological, or technological coordinates; rather, I will insist that we are also talking about *how* (their emphasis) thinking confronts that thematic, what thought has to become in the face of those challenges” (Wolfe, 2010)

To discuss this in more detail, Wolfe is suggesting that posthumanism is the methodological approach by which the human and the machine or technology that they utilize are treated as equal subjects in the interview process. We need to think about not only the device, but the process and culture that surround that device, and how these artifacts are included in the consideration of technology. This concept is expounded upon in Ferrando's *Towards a Posthumanist Methodology,* where he considers to role of artifacts in the interview process, noting that inanimate objects can have a role to play in constructing a complete interview (Ferrando, 2012).

This idea is carried further still. In their work exploring the posthumanist interview process, Lisa Mazzei offers the perspective that the posthumanist interview might be a composition of questions with "lived experiences" that are an assemblage of the human and nonhuman triggered by an agentic clash on the surface with data (Mazzei, 2013). The removal of agency is key to this process. In short, an interviewee's experiences and exposures are relevant only in that they contextualize the artifact at the center of the inquiry (which in our case is TACPS). It is inherent in the adoption of the posthumanist approaches that the interview does not focus on the individual but on the technology at hand and the approaches that surround the technology.

As you will see as we move into the next sections, this principle was inherent in the questions that were composed - questions which decenter the individual and give primacy of place to the technology in question.

Section 2 – Methodology, Study Design, and Hypotheses

Section 2.1 – Methodology and Study Design

This study was conducted in alignment with a project titled "Testing & Evaluation for Autonomous Cyber-Physical Systems: Research and Industry Best Practices" sponsored by the Airforce Institute of Technology (AFIT) in which Dr. Robin Poston served as PI. The interviews were conducted by Dr. Mark Gillenson and the author of this work over a period of one month beginning in March 6th, 2020 and ending April 7th, 2020. In total seven interviews were conducted with testing experts from a variety of private sector and academic settings. These interviews were conducted and recorded via BlueJeans, a video conferencing service subscribed to by the University of Memphis, and the interviews were transcribed by Chaitra Anne, a Graduate Assistant in the Business Information and Technology department. The transcripts were then analyzed and coded using QSR International's NVivo 12 software following guidelines established in two sources which detail application of the technology for qualitative analysis (Bazeley & Jackson, 2013; QSR International Pty Ltd., 2018; Richards, 1999).[15] The transcripts (transcripts are note included to comply with Air Force Institute of Technology (AFIT) study requirements, but may be made available upon request) were used to generate the codes and thematic analysis in NVivo which will be discussed at length later in this work.

Interviewing is a traditional method of qualitative research data collection which typically employs a set of rules that are applied, archived, and adhered to as data is collected in order to achieve uniform standards and practices in constructing the dataset that the analysis is based on (Jamshed, 2014). There are several different types of qualitative interview structures including un-structured, semi-structured, and even more in-depth constructions all of which seek to create a "controlled conversation" that allows for some uniform comparison in order to digest and represent the experiences of subjects under scrutiny (Johnson & Gray, 2010). As David Gray outlines in his work *Doing Research in the Business World*, there are several special

[15] Coding in qualitative research is the process of labeling and organizing your qualitative data to identify different themes and the relationships between them. Coding qualitative research to find common themes and concepts is part of thematic analysis, which is part of qualitative data analysis.

Thematic analysis extracts themes from text by analyzing the word and sentence structure.

considerations which should be given to interviewing employees of companies. Gray highlights how business intelligence concerns can sometimes lead to obfuscation among interview subjects, but that there are several ways the interviewer can create an environment for the interview that encourages participation among professionals who work with trade secrets. Chief among Gray's principles is that the interview be at least semi-structured and that the focus of the interview be restricted to a highly specific topic that would minimize the requirement for specific details about sensitive business intelligence or industry secrets (Gray, 2019). Utilizing Gray's principle in the analysis of the process-as-artifact at the core of this work required that the questions focus on the implementation of testing irrespective of specific technologies or potential industry intelligence that would prevent full details on the processes from being disclosed. At the outset of every interview the emphasis on process and not specific technologies or potentially secret concepts was emphasized. By placing the entire focus on the testing challenges faced by the interview subjects, I was able to dig more deeply into the specific concerns that were at the heart of our structured interview questions. This technique has been used in previous studies that examined qualitative data on testing through interviews with industry professionals (Gillenson, Stafford, & Shi, 2020).

In order to preserve consistency across all the sessions, the interviews were conducted using the BlueJeans virtual platform mentioned earlier. The literature on conducting qualitative interviews via virtual platforms has shown that for geographically disperse groups (O'Lear, 1996) or for structured interviews (Salmons, 2009) an online environment can be preferred. It was fortuitous that we chose this method of conducting interviews as the COVID-19 pandemic set in during the middle of the interview period, forcing online interviews. It has been shown that having all of the interviews consistent in form allowed for better qualitative data analysis

(Salmons, 2014). Literature on qualitative virtual interview also points to the fact that for semi-structured interviews, such as the ones conducted for this study, an online platform can sometimes be preferred as it reduces extraneous factors or environmental concerns (Leech, 2002). We can further point to literature to explain how this virtual approach helps employ a posthumanist approach in the interview process. Although an ideal posthumanist interview would be conducted in the space where the artifacts can be engaged directly (Mazzei, 2013), in this study I am concerned with an artifact-as-process. It has been shown that when considering abstractions or non-corporeal forms in the interview (such as a process), the creation of a "blank space" where the interviewee can project their concept in a uniform way across a space can sometimes be more valuable than when the device is at hand. This projection process allows for language to fill the gaps and can create greater depth in the data (Austrin & Farnsworth, 2005; Oyebode, Patrick, Walker, Campbell, & Powell, 2016).

Previous studies that examined artifact-as-process in the MIS context have emphasized the role of the artifact in mediating communication between the designer and audience, heavily focusing on the exploration of user experience and are an example of *ultimate particular*, or the "unique outcome of an intentional design process which evokes particular emergent properties through the interactions between technical, human, and organizational elements" (McKay, Marshall, & Hirschheim, 2012). In most cases the studies where artifacts are considered, they serve as either theoretical abstractions or are discussed through an analytical lens that disregards the qualitative value of considering the role of such an artifact within the continuum of the larger environment in which they reside. Incorporating these objects into the analysis has been identified as "data assemblage" (Nordstrom, 2015). This concept is where posthumanist approaches most directly influence these studies.

Creating space for the process-as-artifact to be fully integrated into the data assemblage while minimizing the need for references to specific or secret industry technology remained central to the approach as we developed the questions for the semi-structured interviews and began collecting data. Before addressing in greater detail the analysis of the data, I will note briefly here some considerations of certain questions in the interview (for reference please see the full questionnaire, Appendix A). In total there were seven main questions, each with between one and six subparts for consideration. After a brief preliminary session where we asked for any specific questions from the interviewee, the following statement was read:

> "As part of a task for the Air Force Institute of Technology (AFIT) Scientific Test and Analysis Techniques (STAT) Center of Excellence, the University of Memphis is conducting a survey of best practices in government and industry regarding software testing of systems with autonomy. It would be most helpful to our effort to learn your answers to the following questions."

The seven main questions were all focused on major categories of information in TACPS – description of the autonomous cyber-physical system in question, and then information on input data, software, software testing methodologies, software testing execution, software testing standards and measurements, and a final general question. In addition to these questions, a multiple-choice addendum was included with some questions specifically inserted by AFIT. While the questions were fixed, the semi-structured nature of the interview meant that when topics were presented by the subjects that were of note, the interviewers did digress into specific topic areas for further consideration.

Upon completion and transcription of all interviews, I took the recordings from the interview and began to dissect their content. The process of taking recorded interview data and converting it into usable data in a qualitative study is one of the main tenets of the grounded theoretical approach. While this study leverages posthumanist theoretical approaches and considerations in order to remove a human-centric bias in the work, [16] the execution of the study itself and the approach to collecting data relies heavily on the grounded theory methodology for qualitative analysis.[17] Grounded theory is a methodology in the social sciences that involves constructing theories through methodological gathering and analysis of data and relies on inductive reasoning (Charmaz, 2009). Given that a principal outcome from this study is to develop a set of guidelines for TACPS, we need to go into the process with no preconceptions on the topic that would form the guidelines. In a typical study, grounded theory-based research does not utilize hypothetico-deductive approaches, but in this study, I will present a small number of hypotheses that serve to frame the analysis of the data. The precedent for establishing hypotheses in grounded theory-based studies comes from one of the founders of the field, Barney Glaser, who viewed grounded theory as a more laissez-faire type of operation which is inherently

[16] Grounded theory is one of three common approaches to qualitative methodology – the other two being ethnography and phenomenology. The Grounded theory approach works best in the context of this study because grounded theory questions are often open ended "how is xx done/performed/etc?" and can be used to identify the mechanisms and approaches utilized by testers. Phenomenology by contrast, would require a preconceived notion from the researcher (called bracketing) which then could be augmented through research to develop a more representative view (intuiting) and then finally using this data to *analyze and describe* the phenomena being observed. An ethnographic study would be a deep dive to analyze both the vertical and horizontal components of the testing environment and would most likely focus on a specific company in-depth for a longer period of time.

[17] Human-centric bias is quite literally when a theoretical construct or paradigm places the human at the center to the detriment of other considerations. Also termed "anthropocentrism", the concept is at the foundation of most posthumanist theoretical structures. The term has become especially salient with the rise of artificial intelligence and automation, where implicit bias can have amplified impacts on systems design. Accounting for this concept in the consideration of TACPS becomes increasingly critical as autonomous systems take more and more control of systems which can have real impact. As an example of this, we can look at Anupam Chander's "The Racist Algorithm?*" w*hich considers how the implicit bias of systems can be baked into our artificial intelligence structures and systems. The author admonishes that "the turn to algorithmic decision making does not break us free from prejudices" and, especially relevant for the testing audience, "the black boxes of the past may have been analog but they were every bit as obscure as the digital black boxes of today" (Chander, 2016). It is the job of testers to be aware of the risk of implicit bias and ensure that the systems we create do consider these challenges and that is another reason for including posthumanist perspectives in this analysis.

flexible and guided primarily by informants and their socially-constructed realities (Heath & Cowley, 2004).[18] Grounded theory methodological studies should not be so specific in the construction of data that it makes the results time-sensitive or relevant only to a specific context. To further address this issue and ensure a more timeless approach, Alizera Moghaddam proposed an approach to solving the challenges of coding data in grounded theory by laying out some generalized hypotheses that can serve as guideposts, to be proved or disproved, but ensuring that the research questions are considered in full (Moghaddam, 2006).

The removal of implicit bias is a recurring challenge in literature on grounded theory and is also essential to the posthumanist approach. The human-centric bias that the posthumanist seeks to remove is a critical component as I consider the topic of this work – it should be negated where possible. This becomes most salient in the design of the codebook for the data analysis (appendix B). There are typically three levels of coding in qualitative data analysis:

- Open coding - where the researcher begins to segment or divide the data into similar groupings and forms preliminary categories of information about the phenomenon being examined

- Axial coding - following intensive open coding, the researcher begins to bring together the categories he or she has identified into groupings. These groupings resemble themes and are generally new ways of seeing and understanding the phenomenon under study

[18] Informants – in qualitative theoretical approaches, the informant (or key informant) is opposite of the focus group. A key informant is an individual with expert knowledge on a topic while a focus group may be a diverse mix addressing a complex idea. Both can be utilized in qualitative studies, but in this study we only use key informants. The term is used interchangeably with "interviewee" and "interview subject". Glaser used this term specifically, so it is preserved here.

- Selective coding - the researcher organizes and integrates the categories and themes in a way that articulates a coherent understanding or theory of the phenomenon of study. (Glaser & Strauss, 2017).

In order to complete an effective analysis of the data collected in this study, all three levels should be employed. While the codebook for this analytical process as generated by NVivo is included as appendix B, I will not the creation of the codebook as a process in this work. Rather, the analysis and results will emphasize the important trends found throughout the data and the synthesis of these trends into a more cohesive findings and recommendations for industry.

Section 2.2 – Research Questions and Hypotheses

As mentioned in previous sections, I laid the groundwork for the larger study including detailed coverage of both the most relevant literature on the testing of autonomous cyber-physical systems and the qualitative methodological traditions of both the MIS literature and posthumanism. This study will leverage these sources to create a qualitative interview approach that solicits from the interview subjects' answers to the following major research question:

- How are autonomous cyber-physical systems tested?

This question addresses an area of research which literature on the topic has failed to fully explore beyond the sources already discussed. There has not been a survey conducted on the best practices and approaches for testing autonomous systems. In order to fill this gap, I will rely on an approach which contrasts traditional systems testing against autonomous cyber-physical systems testing. I can do this because prior literature has indicated that there is a very real challenge in building and testing these systems which is not entirely addressed by traditional systems testing approaches (Hummel et al., 2018; Ma et al., 2018). What is not clear, and what I believe can best be uncovered through direct survey of testing professionals, is where those differences become most salient. The first hypothesis for this work is as follows:

> *Hypothesis #1: Testing Autonomous Cyber-physical systems requires techniques beyond those employed in testing traditional application software. This leads to new skill and knowledge requirements for the software testers.*

In the analysis of hypothesis #1, I will be heavily relying on the Myer's principles to dissect from the multiple interviews conducted a set of trends and thematic considerations (Myers, 1997b). These will be cross-referenced against traditional testing approaches as indicated in the literature review and the section outlining the definition and background of testing in order to distinguish the unique characteristics of TACPS. The outcome in the analysis will be a set of recommendations based on industry best practices for TACPS.

But there is more to this than simply "best practices". Mark Twain once said, "the less there is to justify a traditional custom, the harder it is to get rid of it" (Twain, 1876). Certainly, this is a sentiment that anyone working in testing should have in the back of their mind, as the landscape that we work in shifts so rapidly. "In testing software there are no "best practices" that we simply must "follow" in order to achieve success" is a fantastic translation of Twain's sentiment into the concerns of the study presented here (Bach, 2003). How do we juxtapose the intended output of a set of recommendations for best practices from the first hypothesis with the sentiment expressed by Bach? I believe that to answer the question fully I must broaden the consideration of the study to think about the variety of inputs which go into the testing process. The second hypothesis deals with the expected inputs and the use of those inputs:

Hypothesis #2: The specific inputs (software, data, etc.) which are utilized in TACPS will be consistently emphasized among participants.

While one participant may discuss sensors and hardware inputs, and others may discuss the role of testing in the development lifecycle, company goals, product specifications, or other considerations, they will all share common concerns which unite their approaches to testing autonomous cyber-physical systems. This hypothesis resonates given that in TACPS there is a diversity of traditions from which developers come to their roles (for example: engineer versus MIS). That we would expect to see commonalities in their specific input concerns is not a new theoretical construct and reflects the principles of the absorptive capacity framework outlined in MIS literature. The absorptive capacity framework is a dynamic capability pertaining to the knowledge creation and utilization that enhances a firm's ability to gain and sustain completive advantage (Zahra & George, 2002). It is the ability for a firm to quite literally absorb knowledge from multiple sources in order to enhance capabilities, and it has been shown in at least a few studies to have ramifications for systems testing (Hadjigeorgiou & Potvin, 2007; Malhotra, Gosain, & Sawy, 2005).

While I am interested in the practical concerns associated with the first research question and hypothesis, there is an opportunity in this study to consider the interdisciplinary nature of testing and the composition of experiences which inform TACPS. As Terry Pratchett once said, "it is important that we know where we come from, because if you do not know where you come from, then you don't know where you are, and if you don't know where you are, you don't know where you're going. And if you don't know where you're going, you're probably going wrong" (Pratchett, 2011). The convergent nature of an emerging discipline means that this work should carefully consider the patchwork of experiences which inform testers and the challenges that they may face - we need to understand where current testing practices came from, so we know where they are going. This principle has been mentioned at several points in earlier discussions of how systems testing shares its roots with engineering. The convergence of these once diverged disciplines in TACPS creates an opportunity for studying how the two concepts are related to one another and where modern TACPS traces its techniques and approaches back to. This leads to the development of subordinate a hypothesis which is examines the question "what are the traditions (cultural, technical, or other) that inform the world view of the testing subject matter?".

This is not a separate research question as it is an extension of the hypotheses stated above. This question is also not explicitly announced in the survey in the way the other hypotheses are but is rather collected through a passive process in the posthumanist data assemblage observational approach. The Myer's Principles show us that there is information which may be presented in the interviews that is beyond the scope of the interview. It is critical that I be able to digest these questions and understand how they impact the work (Myers, 1997b). Because of the highly technical nature of the discussions, it is natural that topics pertaining to other technologies will also be referenced. It is also important to note that the question above is framed in such a way that it is inherently posthumanist – I can consider the tester, the test, and the machine or system being tested in equal measure. Being able to incorporate those themes into the discussion is critical to the analysis of the results. This leads naturally to a third and final the hypothesis:

> *Hypothesis #3: The cultural, technical, developmental, and other related traditions will influence the way in which TACPS evolves.*

In order to answer the question fully I need to understand not only the essential components of the testing procedures, but the more complex network of decisions and background information that inform the decision-making process. This is in line with the discussion earlier in this work which explained the varied traditions which comprise the artifact term that I am addressing. Further, by framing this discussion through a posthumanist perspective, I can cast in sharper relief the process-as-artifact I seek to explain in this study.

Section 3 - Results

Section 3.1 - Introduction

Seven interviews were conducted over approximately a one-month period. One of the seven interviews was discarded because the interviewee was a senior executive who could not provide detailed responses to any of the questions and so could not contribute meaningful data. The transcript of this interview, which we will label #0 for reference along with the six interviews (#1-#6) used in the results presented here (because of agreement participants related to this study, transcripts are not included as an attachment but can be made available upon request). Included in the interviews were four separate companies – four interviews with hardware/software testers from three separate companies and two interviews with university applied research organization members who work with automotive manufacturers in the testing of autonomous cyber-physical systems.

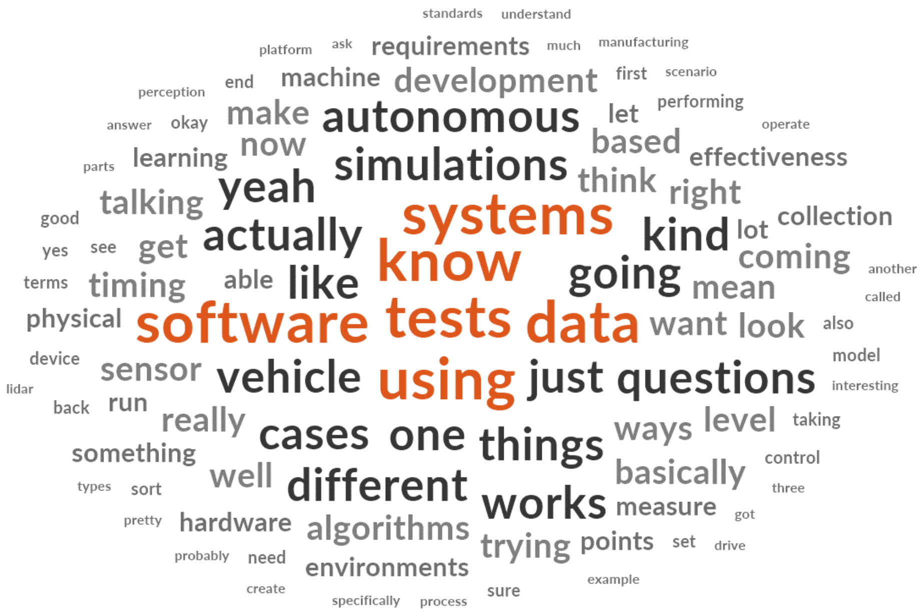


Figure 5 - Word cloud generated from six interviews conducted for the study (generated by NVivo).[19]

Transcription analysis of the data was conducted using NVivo (QSR International Pty Ltd., 2018). Two types of coding analysis were conducted using the platform – thematic coding analysis and sentiment analysis. We will address the results of the two analyses separately, focusing chiefly on the thematic analysis. After dissecting and coding the interviews, the total data output for thematic analysis included seven meta categories, 306 unique coded themes, and 2059 references across the six interviews that were analyzed (refer to Appendix B for codebook of themes). This resulted in a total of eight thematic meta-categories in descending order of prominence by number of codes utilized across all interviews – testing, system, data, software, sensors, level, using, and vehicle (for detailed code count please see table 2). While these meta-categories do exist, the entire data set revolves around testing and so we will use testing as a

[19] Do word clouds really tell us anything? Some would say no (Felix, Franconeri, & Bertini, 2017; Harris, 2011), including me. This word cloud is not being used in analysis, but is a nice inclusion to begin the analysis section.

baseline for the results, and then discuss the other categories as an intersecting group based on their distinct relevance.

Table 2- Coded Meta Themes by number of references in each interview. For specific names of themes see codebook (appendix B)

	data	level	sensors	software	system	testing	using	vehicle	Row total
Interview #1	3	2	4	2	12	18	5	5	51
Interview #2	9	2	4	1	9	1	0	0	26
Interview #3	15	6	5	4	12	29	3	5	79
Interview #4	15	0	5	4	7	8	7	2	48
Interview #5	8	7	3	6	2	15	0	5	46
Interview #6	2	5	2	8	11	22	4	2	56
Column Total	52	22	23	25	53	93	19	19	

3.2 Code Theme Analysis - Testing

It should not be surprising that the meta theme of "testing" is the most prominent both in number of thematic codes and number of mentions in the interviews, comprising as much as 41.57% of references in a single interview. Within this concept we find a diversity of responses that reveal fundamental categories of data which we will explore in the analysis. The most dominant trend in the conversations that were coded for testing concerned the challenge interviewees face in grappling with the transition from traditional testing environments and structures to ones which enable autonomous testing. Therefore, this is emphasized. Other topics and discussions that will feature in the analysis section have also been highlighted in the results.

3.2.1 Physical Testing, Simulation Testing, and Hybrid System Testing

The richest single topic of discussion was the interplay between physical testing and simulation testing and understanding when and why they are used in TACPS. In all interviews both physical testing and simulation testing were utilized, but the degree of integration between the two varied considerably. To help explain this I will frame the discussion with a continuum chart that helps to illustrate the variation in adoption and integration of physical and simulation testing (Figure 6). It should be noted that the differences between testers within this continuum is not a reflection on skillsets or capabilities, but most often is an indication of the different sizes, directives, and functions of the companies and organizations that were interviewed. In some cases, a single interview might contain several examples of testing within an organization, and so some interviews will appear in multiple stages. The continuum presented here is also not meant to serve as analysis but is merely a way to organize the data that was collected.

I have outlined four stages of integrated physical and simulation testing for TACPS, but we did not interview anyone who was not utilizing simulation testing. While there are still testing organizations that do not use simulation testing, the emphasis on cyber-physical systems in this work meant that simulation testing was nearly a prerequisite for testing the technologies in question. The chief reason for this, as revealed in the interviews, was a matter of efficiency - in order to generate the variation and quantity of data required by regulators simulation testing was necessary. "Stage 1" is included here to recognize that testing is an evolutionary process, and that we are only examining a snapshot of a diverse and rapidly changing landscape.

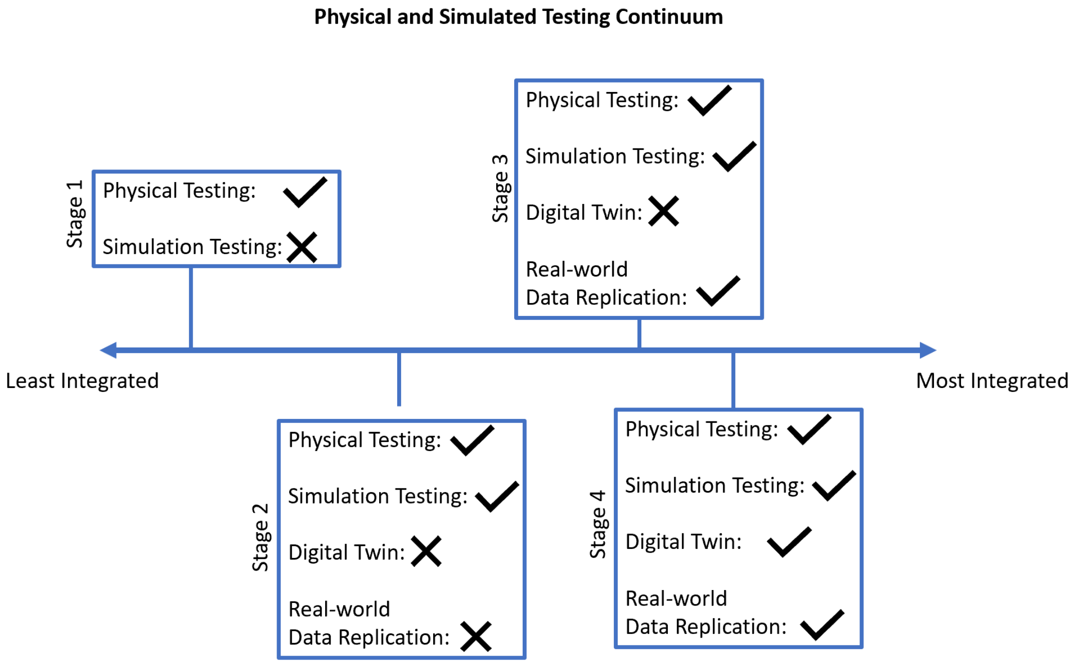


Figure 6 – Based on interview data, this continuum of testing integration is designed to help illustrate the different levels of increased integration of simulation testing which would seem to correspond to level of autonomy and complexity of the cyber-physical system.

Several quotes from the interviews address the interplay between the two types of testing specifically. We can see some examples of testers who have faced real challenges in integrating the two:

> "And the problem is when you touch physical world, it's not as easy (to simulate). You can't do in those (simulation) tests on devices (what you would do) in the physical world. Because you actually need to be able… to control the signals and that's you know, that's doable on a single device." **(Interview #3)**[20]

> "We do physical testing as well. We don't use AI in our testing, we test AI algorithms.

[20] Interviews were transcripted in part by Dragon Software. Formatting has been preserved to maintain vernacular of the participants. In some cases transcription was not clear and so manual transcription override was conducted.

> We can test them off board the vehicle. I guess you consider a simulation because it's offline. It's not part of the full stack, but we can take an AI algorithm and just feed it images then look at how it classifies things on the output and then score them based on how many pixels it got." [21] **(Interview #4)**

> "We use the simulation on the front end basically to help develop the algorithms and so, you know as I mentioned like the machine learning algorithms that are not actually implemented on the platform yet" **(Interview #2)**

In these three examples there is a diverse interpretation of what I have called "Phase One" in the continuum shown above. All three examples do explicitly show the integration of simulation testing, but there is no consensus on how it is implemented. The examples highlighted also show a challenge of addressing TACPS at different scales. The level of integration of automated simulation testing varied depending on the size of the project. At this level, little regulation was present and so the data that was generated was more focused on addressing specific requirements of the system but did not incorporate a deep and iterative testing infrastructure. Frequently a segmented portion of the system would be tested in a simulation but did not incorporate full sensor suite or replication of real-world data. This made substituting changes a challenge as each test was designed to fit the requirements, and not the comprehensive system. Among the interviews conducted, "Stage 2" most often occurred earlier in the development cycle when full scale implementation was either not necessary because that was not the role of the individual or organization, or because the product had not been fully implemented.

[21] Full stack - the entirety of a computer system or application, comprising both the front end and the back end.

“Stage 3” testing integration is differentiated from the prior stage both by the presence of real-world data and the synthesis of multiple component parts in the analysis. The prior examples from “Stage 2” were distinguished by their comparatively isolated approach - testing algorithms in isolation rather than testing algorithms in the system, removing real world data to simplify the test, and removing components from the full stack. At this stage we begin to see a more complex world evolve:

> “We have a simulation of our Hardware-in-Loop solution called [redacted]. [22] [23] It has a hardware-in-the-loop component that's actually generating the simulated environment in real time. And it's feeding that information just as regular sensors would be going into the system. So using the same interfaces that get received by our development system that's running our production software and it's actually you know able to operate and run through a couple different test cases and scenarios before we actually go onto the road. So before we actually go drive the vehicle, we're actually taking, you know, we have a bunch of test cases that are set up with our simulation platform or able to run through.” **(Interview #6)**

> “We actually have a re-simulation system that we can take pre-recorded data that we have taken previously and try to alter it in some ways using deep neural networks (DNN). So there's something more that our research group is working on lately.” **(Interview #4)**

[22] Hardware in the loop (HIL) simulation is a technique that is used in the development and testing of complex real-time embedded systems. HIL simulation provides incorporates the complexity of the environment in which the device will operate. In the case provided here, the environment is an autonomous vehicle, hence the references to going on the road. The complexity of the environment under control is included in test and development by adding a mathematical representation of all related dynamic systems and input variables. These mathematical representations are referred to as the “plant simulation” – a term derived from the fact that HIL was originally designed in a manufacturing environment. The embedded system to be tested interacts with this plant simulation.
[23] Redactions of specific names of platforms have been made per agreements made with industry participants to participate in the survey.

> "How do you measure the effectiveness of the test? Well for instance in simulation we try to validate simulation models with physical testing." **(Interview #5)**

> "Nowadays you have to simulate the controller, the whole vehicle, and the whole behavior of the vehicle, but also everything around the vehicle. So all of a sudden simulation became an extremely powerful tool in how you can deploy just because of any complemented are included in the simulations action" **(Interview #6)**

In these examples of "phase 3" we see the entire system being tested in a comprehensive virtual testing environment. The simulation generates the same variables that the cyber-physical system would experience in the real world, but in a totally virtual environment which allows for testing of components both in isolation and with other factors incorporated, eventually to include the entire functioning system.[24] These tests are more advanced and more complicated than those of "stage 2", and the interviews that were conducted with users of this type of testing shared the characteristic of having fully functional prototypes that were being tested for compliance with standards to ready them for market. These tests did still have limitations that included the generation of enough test case data which led some interview participants to implement deep neural networks to use their decision making and variation capabilities, create derivational randomization in the data so that the machines would have more responsive capabilities (i.e.- allowing the system to virtually drive the same road 10,000 times and introducing random variation to train the system to respond to unpredicted deviations like weather or obstacles). The real limitation of "stage 3" tests, which the use of DNN exemplifies, was that the variable inputs

[24] Other factors which can be incorporated into the system include modified weather conditions, traffic flow rate changes, failure of a component of the system while in operation, lifetime wear from operation usage, etc.

were limited – it was impossible to collect enough real-world data in the timeframe needed to fully test the requirements of the system. This limitation was overcome by using DNN to create rapidly simulated data that would allow for the variability of the real-world anomalies that would be most likely to cause a failure of the system requirements to be tested quickly:

> "you know virtual for example in other studies and to even start doing data collection and analyzing traffic models and based on this analysis traffic models, you how traffic is going to behave. You just don't want to analyze the traffic for a day, you want to do it for 10 years. The real-world data that you want to simulate is not the conditions that happen every day, but the conditions that happen once every ten years." **(Interview #6)**

The goal for these testers was to achieve real-time, real-world conditions and variability, but doing this without real-world testing was a challenge. Real-world testing provides ideal data because it allows for complete sensory data inclusion, but augmentation of a limited set of real-world data iterated upon with machine learning techniques allows that data to be extrapolated upon to build indexed knowledge for the system quickly. According to the interviews, the real challenge of this type of simulated testing is not only creating the simulation but modeling all the inputs for all the variables in the environment and creating the testing structure on top of that:

> "Not only do you have to model the whole simulation platform and the complications of that platform, but then you also have to model the V and V inverse relationship on top of that – it is almost like a toolkit that you model in addition to the other variables. You also have to design in this v and v and automate as many things as possible. You don't want to

write manual test cases." [25] **(Interview #6)**

"When you are emulating a hardware device, and this actually doesn't matter what it is. If you're simulating headlights on a car or the fuel pump or whatever. You can also have bugs in your test devices because to simulate those devices, what I typically do is I just have a microcontroller that acts in their place and this doesn't matter if you're doing it as a single microcontroller or if you have a desktop that basically has a multiplex CAN bus on it you can still screw up the test code." [26] **(Interview #2)**

[25] "In V Model there are some steps or sequences specified which should be followed during performing test approach. Once one step completes we should move to the next step. Test execution sequences are followed in V shape. In software development life cycle, V Model testing should start at the beginning of the project when requirement analysis starts. In V Model project development and testing should go parallel. Verification phase should be carried out from SDLC where validation phase should be carried out from STLC (Software Testing Life Cycle)" (Naveen, 2015)

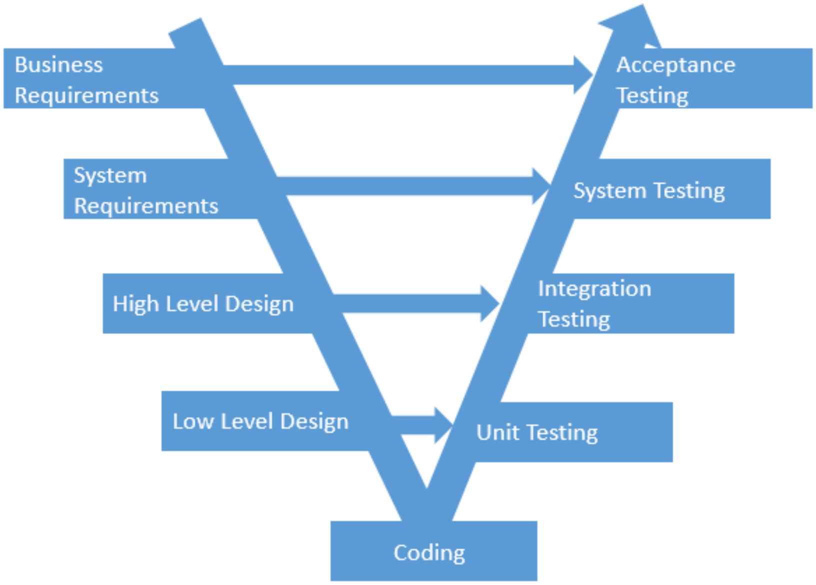


Figure 7 – Illustration of V Model Testing, redrawn from cited source (Naveen, 2015)

[26] Multiplex CAN (Control Area Network) bus (internal communications network) is a method for sending data that does not fit into the 8 byte CAN payload. In some cases, it might be best to send it as multiple messages. One signal in the frame is used as a multiplexer and the remaining payload is interpreted depending on the value of the multiplexer signal.

These attempts to imitate the real world of the cyber-physical systems testing environment are a necessary evil that the testers must wrestle with. Once the model is built, as the previous excerpts state, you will be able to automate the writing of test cases. However, the ideal would be not having to create the simulation in the first place, but instead be able to leverage the best of both worlds – real world and virtual world testing.

There was, however, another version of testing that fully incorporated the advantages of the simulation testing and the accuracy and value of the real-world testing. I have labeled this "stage 4" integration testing, which incorporates comprehensive digital twins into the calculation of TACPS.[27] This was only done by one organization that participated in the survey, and they were focused more on the research and testing of prototypes for industry, not scaled production:

> "One effort were working on right now is creating a digital twin of that environment in this simulation platform that he told you about this morning MAVS ,there so then we can go out and do physical testing and do simulation on nearly identical environment , you know this as close as we can to clean environment" **(Interview #4)**

> "Not in real time the drivers before they go to take a given data collection run. I haven't planned for what they're going to collect and that's documented. And then after the fact the drivers they document any special things, they saw sort of tag that you know set of data along with some things that they observed along the way that's about the extent of

[27] A digital twin is a digital replica of a living or non-living physical entity. Digital twin refers to a digital replica of potential and actual physical assets, processes, people, places, systems and devices that can be used for various purposes. The digital representation provides both the elements and the dynamics of how an Internet of things device operates and lives throughout its life cycle.

> driver input data." (**Interview #3)**

> "Obviously we have the ability to do the simulation, you know with set test cases, but also we actually have the ability to you know, have a human operate the simulated environment that we can actually have them play other actors in the environment potentially or we have the ability to you know, create some Dynamic scenarios as well" (**interview #5**)

In the first two excerpts we see that the digital twin environment is relatively nascent compared to the prior examples, but that it also appears to be a natural evolution of the hybrid models that have proceeded it in prior stages. We see a dynamic data collection model that allows for comprehensive human interaction with no planned test cases established in advance. The real-world road testing and the simulation testing are identical because of the digital twin. By merging these models together, TACPS can incorporate both interactions with real-world variables and simulations of variables that cannot be simulated in the real world. The real-world data can be extrapolated upon. The third excerpt included also shows how a virtual environment with human interaction can also be leveraged to add another dimension to the testing architecture – one that allows rapid prototyping and changes of system requirements. This serves as a half-step between the real world and virtual simulation testing.

3.2.2 - The Role of Neural Networks in Testing

I discussed in detail the role of neural networks in testing literature earlier in the paper. What was uncovered in the interview process was that while artificial intelligence is being utilized nearly universally at this point to do everything from testing algorithms to creating simulation data, the neural network applications are still in the early stages:

> "What is the best strategy to train your DNN is an unanswered question. One perspective is that you use in-vehicle data collection, and you have to travel a number of miles to collect the data. Another perspective is that you use synthetic data meaning for example that you get the data from simulation that you're training on, I would say we are doing both. If you look into the safety standards, they want you to provide proof of how you change your DNN and why do you believe that your DNNs are correct. In order to provide those proofs they want you to have more drive time data than is applicable and so you are forced to create a computational method to (show the proofs)." **(Interview #6)**

> "ya know we're doing that research right now not on board vehicles, but we have our we have some of our data sets now that are labeled and that we were using for developing the neural networks and testing those but not field testing the testing the more like on a lab." **(Interview #3)**

As we see in the comments, one of the chief reasons for a lack of adoption of DNN appears to be in a lack of clarity on strategies to implement DNN. While interview #6 identifies two methods for applying DNN in the testing process, interview #3's discussion seems more focused on testing the applicability of neural networks with data sets generally - they are completely divorced from the platform or the testing structure. Further, interview #6's comment that the when using DNN in testing you need to be able to prove how the DNN functions in the system, which indicates a certain amount of reticence on the part of regulatory bodies to use data generated through these means in testing. This topic is at the cutting edge of the field and is still being discussed in conferences today. A recent paper on this topic in the *ACM Proceedings* reveals that the efficiency of using DNN in testing is still chiefly a cost burden for developers stating, "A challenge is that the testing often needs to produce precise results with a very limited budget for labeling data collected in field" (Li, Z. et al., 2019). It is clear from these conversations that while TACPS is more universal, the neural net integration is still very new.

3.2.3 – Value Systems in the Network

While neural networks are not common, a traditional approach of using value-based ontologies was present in even the most basic testing infrastructure interviews that we conducted.[28] The technique is explained simply here:

> "So a street has a value of plus 1 or minus 1, a river has a value of plus one or minus one and we take all the different factors that are built in and we work with the ecologist. So we work with the other various scientists to understand what impact does a street have…They'll say we looked at these several acres and we realized the street inhibits the movement of the (computer) mouse from x-position to y-position. …So what were able to do is when we attach that value system, we're able to train the system to understand and I'm by no means an expert in this but we're able to train the system to understand those rules based on the actual physical data that is existent. And from there we're able to build the predictive tools that allowed to have that next step of being a useful actionable data." **(Interview #1)**

[28] In Computer Science and Information Science an ontology encompasses a representation, formal naming and definition of the categories, properties and relations between the concepts, data and entities that substantiate one, many or all forms of discourse. More simply, an ontology is a way of showing the properties of a subject area and how they are related, by defining a set of concepts and categories that represent the subject. In this case establishing positive or negative values for objects (streets, streams, etc) is part of the ontology that the company has established to educate the AI in it's systems. This rudimentary approach serves as the backbone of many decision making approaches. It also is where implicit bias can be introduced and can lead to biased decision making approaches. (Iliadis, 2018)

While not explicitly addressed in other interviews the universality of the technique and the description of other platforms means that we can deduce that a similar approach is used in other testing organizations that were interviewed for this study. This is important to the posthumanist analysis later in the work.

3.2.4 – Test Cases Discussion

Another major trend in the data is the discussion of specific challenges associated with developing test cases associated with TACPS. While test cases are inherently part of any testing process, the discussion of the role of test cases in TACPS had a unique perspective that emphasized the capacity for leveraging the large data sets to the advantage of test case development:

> "The nature of the test cases, our test Matrix tends to get really big because we consider factors such as day vs night, raining, snowing, sunny, afternoon, hard shadows, morning or overcast with soft shadows. The nature of our test cases are primarily environmental, you know, what type of plants? what type of terrain ?color palettes generally around the vehicle and other environmental factors like time of day and weather and even how long it's been since I lost range of those kind of things." (**Interview #3)**

> "I don't really think there's a one-size-fits-all metric (for measuring test success), it sort of depends on the mission objective." (**Interview #3**)

In these two examples we see a key feature of the interviews which was the size of the data set, but more importantly for the test cases, the variability of the data in the test case development. Because of the sensor data and the comprehensive nature of the data collection (whether in virtual environment, real world, or hybrid) and the diversity of the objectives that can be measured, the testers are not limited in the type of case studies the develop. Once you have established that any test is technically possible, the execution then becomes critical:

> There's nothing that stops you from using machine learning to design better test cases or confirmed retirement to create smart randoms who created, you know, kind of intelligent feedback in your test system. We're doing this on some levels." (**Interview #6**)
>
> "The scenario would be a test case depending on how you would design the modulations of the scenario. Depends on what objective you are looking for." **(Interview #6)**
>
> "So let me Circle back real quick on the isolating individual test cases. I mean I actually do think that's really relevant sense to the way that we're doing things because what we're doing is basically we're keeping the perception and the path planning and the vehicle control algorithms completely separate, isolated and we are able to test each of those subsystems in isolation and we're doing that again from us from a software testing point of view" (**Interview #4**)

Here we see two examples of how this diverse dataset is employed to create test cases. In the first two quotes the interviewee uses the outputs of the test against an established standard (say a safety standard) and using machine learning to build better test cases. They also note that it is the scenario which determines the test case, a comment that is upheld by the comments about the size of the data being able to create test cases that are scenario and mission specific. In the quote from Interview #4 we see a different approach to building test cases employed. Where Interview #6 uses the full system in testing and then leverages machine learning to build test cases to help improve efficiency, Interview #4 creates individual test cases by isolating subsystems and testing them individually. This method is less complicated but also reduces the efficacy of the test against external standards. This difference further emphasizes that it depends on the goals of the organization as to how test cases are approached. While interview #6 is more focused on going to market, #3 and #4 are more focused on testing prototypes for efficacy. This is born out again in this quote:

> “I'm failing on this random spot for example, and I'm failing with the tolerance of let's say 90 percent to 95 percent of my test cases are failing there so there is something going wrong there, which means the engine would create new test cases for you” (**Interview #6**)

Only the interviewees concerned with market-readiness spent significant time concerned with failure tolerance against a predetermined specification.

3.2.5 Discussions of Testing Culture

It was intentional in the interview process not to directly ask the subjects about the role of testing culture. Rather I will comment here on observations related to this topic that seemed to resonate throughout the conversations about testing culture and the execution of the work of the testers.

> “The problem is it depends on what I'm working on. I can probably use Agile methods for (test jigs) because I'm not touching Hardware that much. When you start playing with Hardware it's harder. It's harder to make it work in a purely Agile type environment … it's harder to do Agile style development including testing when you're doing deeply embedded system… it's a kind of a hybrid.” **(Interview #2)**

> “I would say we've been (doing exploratory testing) and it doesn't seem that there are industry standards out there for at least these types of vehicles - telling you exactly how

> you define success or what you measure to compare two different algorithms. The on-road industry tends to use engagements per thousand miles as a metric - how many times does the safety driver have to take over or how regularly does that happen? It’s generally been decided that's not a great way to do it. It doesn't really tell you a lot, but I think some companies still track and measure that as a primary metric. We generally have been looking at average speed, I would say, if we have a course we set up a course to test the vehicle. We want to evaluate two different algorithms of the same vehicle driving on that course, we can set one algorithm to allow the vehicle to hit any obstacles and this one to provided an average speed of 12 miles per hour while this one provides an average speed of 20 miles per hour and that second one would be a little bit better. There tends to be other differentiating factors though like for instance, there may be one algorithm that supports reverse. I mean it's a vehicle so if it tries one path and gets stuck that can it back its way out and try a different one and some algorithms allow that and others don't, so you tend to find more functional differences in how they work than just purely ,this one lets you go 12 miles an hour versus 20 or whatever, so that you hit on a really challenging area that we and others. I think you're just trying to find answers to.” **(Interview #3)**

This first statement reflects on the discussion earlier in the paper on the divergence and then convergence of Management Information Systems and Engineering testing approaches. We see this manifested here, with different types of testing and development requiring different testing methodologies and approaches. The subject says that using Agile on deeply embedded systems is not practical, which would anecdotally at least suggest that in TACPS Agile has no real role. In the second example we see a walkthrough of the exploratory testing methodology as understood by one of the other interviewees. Here we see that that the interpretation of the exploratory component is very similar to A/B testing (or split-run testing) and would most generically be called iterative design.[29] What makes this especially interesting is that A/B is most often used in user experience testing. In this case we have human users, but we also have AI users. Although there is a safety driver in the example provided, the parameters of the algorithm are being tested which has interesting ramifications from a posthumanist perspective.

Apart from these relatively few examples, the interviewees did not explicitly address cultural challenges in conducting testing of autonomous cyber-physical systems. The discussion of V and inverse V testing mentioned in a prior section does indicate that mixed methods approaches in testing were incorporated. I will also comment on the fact that in every interview the subjects consistently highlighted the group over the individual. Saying “we” instead of “I” is reflective of the cultures that are inherent in the type of work being done. Teams of collaborators are essential to execute these testing requirements, and that was reflected in all interviews.

3.2.6 – Exploratory Testing

One of the initial interest areas of the AFIT survey was to better understand the concept of exploratory testing and how it is implemented in industrial practices. The term exploratory testing was first used in 1984 by Cem Kaner, defining the concept as “a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the quality of his/her work by treating test-related learning, test design, test execution,

[29] A/B Testing is a user experience research methodology. A/B tests consist of a randomized experiment with two variants, A and B. It includes application of statistical hypothesis testing or "two-sample hypothesis testing" as used in the field of statistics. A/B testing is a way to compare two versions of a single variable, typically by testing a subject's response to variant A against variant B, and determining which of the two variants is more effective.

Iterative design is a design methodology based on a cyclic process of prototyping, testing, analyzing, and refining a product or process. Based on the results of testing the most recent iteration of a design, changes and refinements are made. This process is intended to ultimately improve the quality and functionality of a design. In iterative design, interaction with the designed system is used as a form of research for informing and evolving a project, as successive versions, or iterations of a design are implemented and test result interpretation as mutually supportive activities that run in parallel throughout the project" (Kaner, 2008). In every question we asked whether exploratory testing was conducted, and for the most part this was not done or was confused with another testing mechanism:

> "So in that sense (exploratory testing), I would say it's very aspirational. I don't think we're doing this. I would like to do it, but I don't think we are at this stage yet. Now what looks, it looks like is it is very manual to me – changing the lights, the weather and all this. Well obviously, we are doing this but I would call it more like directed test and maybe random parameterization test." **(Interview #6)**

As Kaner says in the work cited, "all testers do exploratory testing. Some do it more deliberately and in more intentionally skilled ways." This seems to reflect the conversations about testing culture mentioned above. That Kaner discusses a variety of techniques for exploratory testing including scenario testing, specification-based testing, and risk-based testing further supports the idea that exploratory testing is a natural extension of the testing culture which is developed in a team environment and the iterative testing necessary on any project.

3.3 – Other Meta-Category Results

As mentioned in the introduction to the section, testing is the overwhelming thematic focus of this data set, and all eight meta-categories are closely related to the major theme of testing. However, there is some information that remains distinct enough from the discussion of testing that it should be examined on its own. These discussions are not as lengthy as the testing focused discussion, and so I am including them in a single section of the work.

3.3.1 – System Meta Category

Because of the nature of the work being performed all the systems were comprised of (in part or entirely) software systems, and so these two categories have been combined in this analysis as they are not easily parsed separately. The data reveals that TACPS is never done on a single system. While some interviewees tested the entire system in situ and others chose to test individual components, all of them were dealing with a complex network of systems that needed to be reconciled in any test. It is important to reconcile the complexity of these systems in order to understand how testing is implemented:

> "If I'm not then the system level design team is going to take care of making sure that there is a redundant system that is going to have this corner case and this redundant system is always going to work. So basically you no longer thinking about unit testing as you know your end solution you investing is just the beginning is like the Pandora Box from now on you're going to be dealing with more and more Complexity going up there v-model." **(Interview #6)**

> "But I think this is what they mean at the abstraction of a now at the concrete level, you know that you're looking for the margins tolerance and keep you guys - obviously you are at the end of the day, but there will be a little bit more here because you are here as well interested in the end to end testing. So you're interested in the mission. How is the mission going for your entire CPS system. And how do you evaluate this mission to the evaluator for the mission is going to be different. This is gonna be That's different. It's just going to be extended in comparison to what you used to." **(Interview #6)**

One interviewee highlighted the fact that in building the simulation for the system it is possible to combine signals in the system in a way which would not otherwise be possible in traditional testing approaches:

> There's so many signals, but you have to deal with combination. How are you going to combine the signals? So it's not only about how you selected test data for specific Quality

> roof example for velocity. Let's say this for velocity. I'm going to take between 0 and 70 miles per hour and then for breaking I'm going to take the street value either breaking or no breaking. Otherwise, what I'm really interested in are multiple other signals that are influencing the vehicle Behavior and the traffic agents around me and all this we have to go to very clever combination in on the scenario cure and in order for you to be able to analyze the scenario and evaluate the scenario. **(Interview #3**)

Another example of this is one that uses humans as the secondary system:

> "… we also have another stack which is called [redacted] which is basically an in-cab and monitoring and external monitoring system. That's a non-autonomous system, but actually acts as a safety system to help prevent potential safety situations. So pretty operating the vehicle with some of the sensors but not actually autonomously operating system. (**Interview #3)**
>
> (Does that mean is a driver driving the vehicle?) Yes. Yeah eventually could be a driver that's fully in control of the vehicle. But since we're able to perceive potentially like someone running a red light that maybe the driver didn't notice we can actually take some, you know, proactive steps to prevent certain issues that can happen." (**Interview #4)**

This illustrates not only that the systems testers must consider the objectives of the systems that they are designing, but also that they must build additional interim layers of systems to help with training the whole system. We see this further reflected in the quotes included in the prior section discussing system level design and the discussion of test cases. In both we see that the TACPS requires a rich fabric of layered tests – some not directly focused on the objectives of the system being developed or tested - to achieve the desired output. This means that no system can be measured in isolation when considering the product or final total output of the cyber-physical system.

3.3.2 – Data Meta Category

There were a total of 52 references to data coded in the dataset across 37 terms (see table 3). As the majority of the references to sensors was closely related to this category (i.e. – "sensor data", "lidar data", etc.) I have combined their analysis. The diversity of these terms, including everything from training data, to generic sensor data, to the data collection process and many other points in between is an indication of the breadth of the role of data in the testing process.

Table 3 – coded reference metadata for "data metacategory"

camera data	objects data	useful actionable data	geometric data
clear data	physical data	video data right	lidar data
cloud data	radar data	in-house data sets	manufacturer spec data
data collection run	raw data	raw data stream	map data
data collection side	raw image data	recording data	numeric data
data flight	re-simulated data	sensor data	test data
data scrub	streaming input data	streaming data	data collection
direct data	selected test data	streaming input data	
discrete data packets	several data collection	weather data	
driver input data	training data	smart connection level data	

According to the interviewees, there is no way to conduct these tests without good data and the diversity of data sources. Earlier in the section 3.2.1 I provided an example of data and the challenge and opportunity that data in these scenarios holds:

> "…you know virtual for example in other studies and to even start doing data collection and analyzing traffic models and based on this analysis traffic models, you how traffic is going to behave. You just don't want to analyze the traffic for a day, you want to do it for 10 years. The real-world data that you want to simulate is not the conditions that happen every day, but the conditions that happen once every ten years." **(Interview #6)**

From the perspective of data analysis, we see here that the simulation allows for the extrapolation of data from known deviations or potential deviations based on real world settings. The tester is saying "it is dangerous and inefficient to wait for a traffic accident or snowstorm, so let's reproduce those circumstances in the data and see how the system reacts in simulation". This data is also being augmented with multiple types of data streams collecting on a single test. This means not only different types of sensors, but multiple input parameters:

> "Like I talked about we're going to do a tele-op or even just the driven run or several data collection runs on our test course where we collected data and process the date exactly as we would in our economy, but we have a driver with control of the vehicle that we can go back to and look and say okay, this is what this is what it would have done and you know, we're comfortable with that or no." (**Interview #3)**

This type of data collection, and more importantly the reliance on humans as a failsafe, is a major ramification of CPS which I will be able to discuss more in the analysis.

This type of data collection, and more importantly the reliance on humans as a failsafe, is a major ramification of the system which we will be able to discuss more in our analysis.

3.4 - Sentiment Data

Sentiment analysis allows us to quantify perception of a given topic by measuring key words throughout the document that have specific weights and values (i.e. “hate” = strongly negative, “love” = strongly positive) (Hai-Jew, 2017). While we would not expect the topic of these interviews to be an overly sentiment-laden discussion as it is a highly technical topic, it is useful to include the measurement as a comparative analysis between specific interviews. To do this, I adopted the quantification mechanism used by Mike Thelwall and others (Paltoglou & Thelwall, 2010). Essentially this technique uses valuation (-2=very negative, -1=moderately negative, 1=moderately positive, 2=very positive) to weigh the values of the sentiment keywords across the entire interview. This establishes a net score for the interview with lower numbers being negative sentiment interviews and higher numbers being more positive sentiment interviews. You can see the total scores for each of the six included interviews in table four.

Table 4 – Sentiment Data Coding Per Interview

	Very negative	Moderately negative	Moderately positive	Very positive	Net Sentiment Score
interview #1	5	13	18	9	13
interview #2	7	9	21	14	26
interview #3	15	19	32	18	19
interview #4	0	8	5	4	5
interview #5	4	5	14	9	19
interview #6	14	23	25	6	-14

Even with a small dataset we can learn something from this information. First, we know that for these numbers the standard deviation was 14. What is noteworthy in this data is the extremes -the interview which skewed most heavily towards negative was #6, which was dealing most directly with TACPS for autonomous vehicles and DNN. The most positive interview was #2, which was a small startup creating a prototype device. The variation between them should be noted as it is a possible indicator of several challenges and the various phases of development and scales of operation that the interviewees were operating in.

Section 4 – Analysis and Discussion, Summarized Findings, Challenges, and Best Practices Recommendations

4.1 - Introduction

At this point, it would be useful to summarize where we have been to know where we are going next. In the beginning I discussed the concept of process-as-artifact (or process improvement) was at the root of MIS literature (Ackoff, 1967b; Boland Jr, 1978; Harrington, 1994). Then I proceeded to discuss how one type of process-as-artifact, testing of the software in autonomous cyber-physical systems (TACPS), which is not a topic covered in depth by current scholarship. I made sure to provide you with a reasonably clear understanding of what each of the terms that comprise TACPS means and where those terms are rooted. I discussed briefly the history of systems testing and the relationship to, divergence from, and re-convergence with engineering testing principles required by cyber-physical systems. I then dove into the specific literature that has focused on TACPS including some cutting-edge approaches like applying neural networks to help provide background to the reader as I moved towards the data that was collected. Before we got to the actual data though, I laid the methodological approach that was employed which featured a discussion of the Myer's principles and their application to the interviews that were conducted (Myers, 1997a; Myers, 1997b). I also outlined a discussion of posthumanist approaches which I do intend to employ where possible in this section. This brings us up to speed, except for addressing the hypotheses which will be covered in this section.

After analyzing the data that was presented in the coded research discussed in the prior section, there are several interesting topics for further discussion. I will frame this discussion, in the context of answering the three hypotheses proposed earlier in the work. Much of the discussion can be addressed here. Following this discussion, I will lay out summarized significant findings, identified challenges, and finally I will present some recommended best practices for industry professionals.

4.2 – Hypothesis #1: Testing Autonomous Cyber-physical systems requires techniques beyond those employed in testing traditional application software. This leads to new skill and knowledge requirements for the software testers.

During the literature review for this work, I discussed several distinguishing features of cyber-physical systems testing that should be considered. There were four points that were within those works that were highlighted and that I will restate here:

- the presence of machine learning and artificial intelligence in the system functions but also that this technology can be leveraged in the testing of these systems (see section on *DeepTest)*
- the complexity of requirements, fault injection, and non-determinism in a system with the "driver out of the loop" (see section on *Challenges in Autonomous Vehicle Testing and Validation)*
- the "stateful" and "temporal" operations for an autonomous cyber-physical system moving through and responding to its environment (see section on *Robustness Testing in Autonomy Software)*
- project management in a large multisystem project and team coordination (see section on *A Collection of Software Engineering Challenges in Big Data Systems)*

Because these points were laid out in prior works, it is established that autonomous cyber-physical systems are substantially different than their non-autonomous and non-cyber-physical counterparts. However, fundamental differences in the technologies did not necessarily precipitate differences in the skills, approaches, or knowledge of systems testing that was required in order to perform TACPS, and this was one question I sought to answer through interviews with experts.

Through these conversations there were major trends that revealed that TACPS requires fundamentally different approaches to testing which align with the differences in capabilities of these technologies identified in the literature. These differences stem both from the novelty of autonomous technologies, the capabilities of these systems, and the burden of requirements that is placed on developers. In short, these major differences that confirm the first hypothesis to be true can be summarized in three major categories: safety, simulation, and test design.

4.2.1 – Safety

In every interview conducted the topic of safety was central to the conversation and was frequently discussed in the context of both external regulatory requirements and self-imposed development requirements to place limitations on the system, as well as concerns stemming from the deployment of these technologies in the real world. It is not the subject of this study to analyze the safety requirements, but their impact on testing forces the creation of modified solutions which introduce approaches to TACPS that would not be found in traditional systems testing structures. Examples of this include:

- creating simulated and hybrid testing environments (which I will discuss more in the next section) that perform complimentary and intersecting functions, [30]
- the inclusion of systems requirements or testing protocols specifically designed to limit performance within the expectations of human experience,
- testing algorithms and other component parts in isolation because of a danger to or complexity of the larger system.

This creates entire secondary systems to address safety concerns that would not be integral to the primary system design and leveraging ANN and other neural net technologies to create simulated data for external safety review requirements.

Safety is crucial to the development of any system, and so there is no novelty in addressing safety concerns in systems testing. The distinguishing characteristic in these examples, and the reason it is worth highlighting here, is in the novelty of the application of the system itself – safety standards for autonomous systems are only now being established and based on the interviews conducted there seems to be a constantly shifting expectation of system capabilities.[31] This results in systems testing protocols which are dynamic and can quickly adapt test case scenarios to meet new standards and requirements. This is compounded by a lack of general understanding among not only the general public but the industry regulators as to how

[30] This should be distinguished from the role of simulation testing as is normally understood in non-autonomous cyber-physical systems. As an example, the avionics industry uses simulation testing to measure the affects of new or improved parts by inputting them into a simulation which is then tested by a human pilot or testing personnel (Shen & Zhai, 2017; Soneson, Horn, & Zheng, 2016). However, in the examples provided in the interviews in addition to this traditional approach, we also have fully automated pilots, we have ANN-driven data set augmentation, we have human-based foundational data set creation through driving that is then augmented through simulation and several more. This understanding of the role of simulation merges several different approaches into a single practice.

[31] For example, ISO standards for autonomous vehicles are still under development as of this writing (May 2020) and have been for at least a year. Currently the industry is applying ISO 26262 which is really meant specifically for factory non-autonomous robotics, and so the effectiveness for TACPS is limited (Tabani, Kosmidis, Abella, Cazorla, & Bernat, 2019).

these technologies operate (or should operate) which results in a higher than normal visibility for failures.[32] Testers of these systems must therefore have a higher-than normal consideration of safety.

4.2.2 – Simulation

As I discussed earlier in this work, the role of simulation in TACPS is part of a larger continuum, with different degrees of simulation being incorporated in different phases. In answer to the hypothesis above, I will not again belabor those points referenced previously. Instead, I will highlight what is truly novel in the discussion of simulation – the intermingling of three different simulation approaches for TACPS that do not appear to exist in current literature on the topic.[33] To outline this explicitly, the interviews discussed three types of simulation deployment in testing cyber-physical autonomous systems – real world scenarios, virtual environments, and neural networks (see Table 5 for definitions). Not all testing that employed simulation used all three types.

Table 5 – Descriptions of the three types of Simulation Testing Used in TACPS

Real World Scenario	using a prototype vehicle equipped with sensor arrays, the vehicle is driven by a human operator to collect a dataset. The human may be a ride-along (not driving)

[32] Autonomous vehicles especially are subject to this concern. For example, it is without fail that any catastrophic system failure (a wreck, loss of life, or other accident) is made a prominent media item (Koopman, Ferrell, Fratrik, & Wagner, 2019). This is not a simple topic, and as Liza Dixon discusses in her work "Autonowashing: The Greenwashing of Vehicle Automation" the expectations of the vehicles established by the corporations is at least in part responsible for a magnified perception of their failure (Dixon, 2020). The role of the tester

[33] Current literature on simulation in software testing focuses on one component of simulation deployment and not the interplay of all three identified here (Hadjigeorgiou & Potvin, 2007; Pei et al., 2017; Shen & Zhai, 2017; Soneson et al., 2016; Tian, Pei, Jana, & Ray, 2018b)

	or driving the vehicle depending on the stage of development.
Virtual Environment	a digital environment is developed and the system (either a modulated piece or the entire system) can be interacted with in the environment.
Artificial Neural Networks	utilization of an ANN (most often a collection of digital neurons, connections and weights, and propagation functions) trained to perform specific problem-solving solutions for a given task or set of tasks.

I do not believe that the discussions of the application of these simulation approaces is applicable only to TACPS. Because this subject is by its nature at the cutting edge of testing approaches, and given the discussion earlier about the growing prominence of these types of systems in our daily lives, it seems fitting to highlight that the novelty of how simulations are applied here has resonance for all of testing. I find it most helpful when considering the intersection of these three types of testing to look at them as a Venn-diagram because a testing group could use all three concurrently or use combinations of each interchangeably. Based on the interviews, it seems that only when all three are used concomitantly do digital twins (discussed above) become an active part of the digital testing infrastructure. Other combinations can be used effectively and were deployed by different organizations that were interviewed.

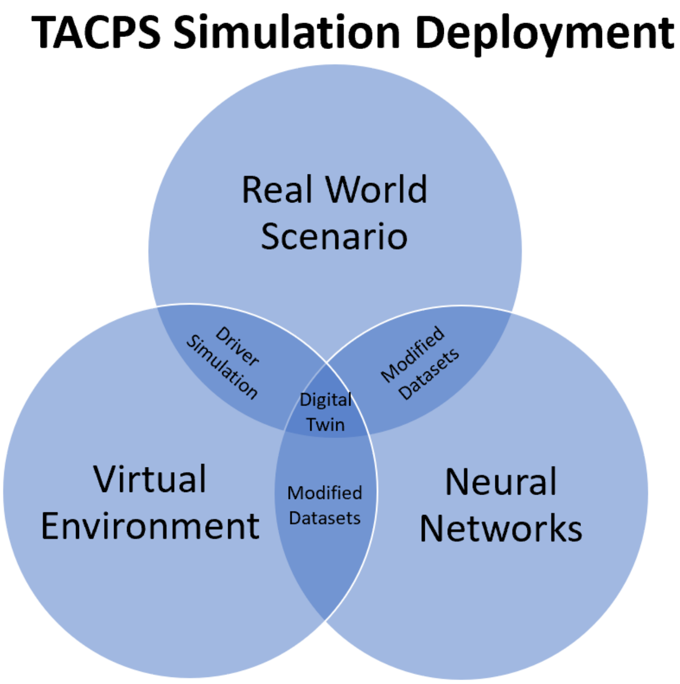


Figure 8 -TACPS Simulation Deployment. Here we see outlined the three types of simulation deployed in TACPS and the intersections between the types.[34]

We mentioned earlier in this work that the posthumanist perspective declares that the role of the human and non-human (to borrow from Latour) is subjective, and for the purposes of the work of the qualitative analyst should not serve as a clear distinction in considering the data (Latour, 2000). Helpfully, the subjects of the interviews have illustrated this through their discussion of simulation in testing. The discussion of all three types of simulation testing place

[34] *These include:*

- Modified Datasets (virtual environment-neural networks) – data from the virtual environment is used to modify the dataset used by the neural network. Augmentations to these datasets are most often made by AI
- Modified Datasets (real world scenario-neural networks)- data from the real-world scenarios is provided to the neural network. This dataset may be traffic or weather data, input data from human driver simulations, or other data generated "in the wild" and is used to inform decision engines in neural networks
- Driver Simulation – a human interacting with a virtual simulation generates this type of intersectional data
- Digital Twin – This intersection combines all of the other types plus adds the total simulation of the real-world conditions so that systems used in real world testing are fully simulated in the virtual environment. The digital twin will perform more rapid permutations on the data than real world testing could perform in a reasonable amount of time.

equal emphasis on the roles of virtual environments, real world scenarios, and neural networks. In all three cases they are seen to share different but equal values in the testing process. Further, and as a most exemplary execution of a posthumanist approach, these three simulations can interact seamlessly and (apart from their initial development) can adapt and modify the test cases based on iterative feedback from the system. Humans do not need to be present for testing to occur in these simulations, except where they are bring intrinsic value not represented by the other components – all three simulation types have equal prominence. Posthumanist literature that focuses on technology often cites a discomfort or distrust that results in a measured or cautious utilization of the technologies (Hayles, 2008). The data assemblage that is the interaction of the testers with the test simulations combined with the safety concerns which stem from an inherent systemic mistrust of the application of AI in situations where human life could be at risk (such as autonomous vehicles) further highlights the posthumanist framework through which we can view this unique intersectional simulated testing environment.

It is important to identify not only that the three types of simulation can interact autonomously, but that at the respective intersecting points we see different types of interactions. Neural networks, when interacting with another single type of testing simulation, manifest that interaction as modifications in datasets. For example, if the neural network is enacting a set of decisions either in a virtual or real-world test case, the feedback and knowledge gained manifests as changes in the database. These changes are then absorbed and become part of the knowledge base that the system uses in future testing. A real-world scenario interacting with a virtual environment is how those interviewed would describe a virtual simulation with real world data inclusion. This could be a human test subject, or data collected from real world scenarios, that is incorporated into or interacts with the virtual environment. Where in the case of autonomous vehicles this can be a human driver in a virtual world, it can also be a virtual driver in the human world. This is distinguished from a neural net approach, as some of the examples we see in the interviews are applying real world simulation data to virtual models, especially in the case of prototype development.

While this combinatory approach that inately leverages a posthumanist mindset is a fascinating part of this study, it is in the potential value of the digital twin in the simulation triangle that we see a real impact on the testing landscape. The digital twin sits at the crossroads of all three types of testing simulation and requires all three to be executed. The digital twin provides a fully structured virtual copy of the real-world simulation environment and allows for the ingestion of both real-world simulation data, virtualized simulation data, and interaction in both the development of test cases and also in the modification of the sensor inputs by the neural network. It is an immensely complicated test matrix, with multiple layers of systems – some of which are solely built to help conduct the test of the system itself. The system is fully immersed in the posthumanist approach, with the system relegating the human operator to one of several input mechanisms.

4.2.3 – Test Case Design

Although not highlighted explicitly in the results section, the data discussed reflects that test cases follow the same continuum as simulation. In the earlier discussion of test cases, I showed that the most complicated test matrices and integrations of automated testing coincided with the systems which test in situ. Because examining in detail the composition of individual test cases was not part of this study, I cannot provide evidence that details the variation in test-cases between those testing systems that isolate versus those that test the components in situ. However, a paper published in *ACS/IEEE Proceedings* did outline how component testing can be utilized to produce efficiencies in the test case development that would necessarily reduce the number of generated test cases required (Beydeda & Gruhn, 2001). Relevant to this discussion are a series of patent filings I found which examined methods and systems for isolating software components (Apuzzo, Marino, Hoskins, Race, & Suri, 2005; Lopian, 2013).These patents are particularly interesting because of how they attempt to structure the system for test case design. It is clear that when we compare these models for system design against the neural networks, that the neural networks are necessarily more powerful (power meaning number of test cases generated) but require a great deal more computational space and deeper understanding of technologies which are not widely adopted by testing organizations. It is apparent from the relatively light use of ANN among the interviews that we are only now reaching a tipping point where the value of ANN overcomes the efficiency of traditional component testing in most circumstances. It is worth further study to examine the transition of these testing architectures as they adopt neural networks, especially for larger industry operations.

For their part, test cases in TACPS at least anecdotally appear to resemble the test cases of traditional systems testing structures. Where they diverged was in the quantity and detail of the tests that were performed. Especially for the vehicle testers, the apparent granularity of the test cases (weather pattern changes for example) meant that automation of the test case development was critical. Utilizing AI and machine learning was alluded to present interesting results, but further study of the specific variation in the test cases would be required to provide a data-driven answer as to whether the differences were measurable.

4.3- Hypothesis #2: *The nature of ACPS and its control software requires new techniques to handle real-time, streaming sensor data and the decision making using this data.*

The emphasis in the hypothesis, as discussed in the earlier section where it was introduced, is on the consistency of emphasis of the types of data and the way that the data is utilized by interview subjects. Emphasis in this case means that these inputs are vital, unique, or distinguishing in the application of TACPS versus traditional testing approaches. In order to prove the hypothesis, certain types of inputs would need to be consistently emphasized. Given that software testers can come from a variety of backgrounds and experiences, a confluence of specific inputs would shed light on the common values and importance of key components. I found that we can prove this to an extent, but the common inputs that were shared were not necessarily a specific software or data source, but rather the understanding of how these inputs were managed in testing the larger system. As an example, we see numerous discussions of sensors throughout the interviews. Included in table three is a long list of data types that would be generated by external sensors (lidar data, geometric data, camera data, radar data, etc.) but fundamentally there was no novelty or unique extraction of data that would be any different from an analog system utilizing the same sensor sets. The sensor outputs are consistent regardless of the system in which they are utilized.

Where the second hypothesis is more effective is in the discussions of the divergent concerns between the three types of simulation testing. This difference is highlighted in the quotes from interviewees and discussion throughout the work, that circuitously expose that within TACPS there exists really two sets of testing standards – one for simulation testing and one for real world testing. This is observed in comments where testers recognize that they have different data sets that they use for testing in the different environments. It is also observed in the desire for the creation of digital twins in testing (see Figure 8 and the subsequent discussion), and the fact that exploratory testing is aspirational but remains beyond the integration of the testing infrastructures in the interviews. The concerns surrounding the challenge of this divergence was a consistent discussion across all interviews and seems to reveal a close interconnectedness with the findings that prove the first hypothesis.

Based on the interviews that were conducted I can conclude that the specific data sources (sensors, etc.) used in TACPS are not fundamentally different from one interview to the next and their emphasis was not remarkable. There was no discussion of a single input that was critical to the discussion. All of them used roughly the same types of systems and the specific inputs were not heavily emphasized in the conversations (see table 3). What was heavily emphasized, and was consistent across all the interviews, was the challenge of reconciling the three different types of TACPS simulation deployment. This deserves further study as we did not explicitly target the challenges of integrating the three, but it is clear from the interviews that these challenges are uniformly difficult to overcome.

4.4 - Hypothesis #3: Software testers will have to learn new skills and adjust to a testing environment that is different from the one they are used to.

This hypothesis, much as the last hypothesis, finds its resolution stemming from the discussion of hypothesis #1. Earlier in this work I discussed how TACPS is a convergence of many different traditions (engineering, software and management information systems, HCI, etc.), and at the outset of the work I alluded to the cultural impact that these autonomous systems can have, their pervasiveness in our daily lives that extends deeper than is often realized or considered among the general population. Yet the seamless operation of autonomous systems - from smart home devices to autonomous vehicles – is due in large part to rigorous testing. As we become more reliant on these systems for increasingly vital components of our lives and thus expose ourselves to risks and potential personal harm in the process, understanding the values that inform how tests are constructed is critical. If we fail to ask the right questions or create the test cases that fully simulate the environments and uses of the systems being designed, then failure is more likely.[35] Hypothesis #3 also expresses a posthumanist concern around the way in which these interactions occur which reflects the discussion of hypothesis #1. We need to know not only the concerns of the psychology of testing, but also the interaction between the components themselves and the testers and operators. I do not feel we have enough data to fully explore the rich potential for analysis which the third hypothesis entails. However, I will provide a discussion of what I do believe we can determine from the study above.

As I mentioned in the results, there was little explicit discussion of cultural traditions around testing among the interviews beyond the referenced team mentality mentioned above in the discussion of pronoun placement. I do not believe this is sufficient evidence from which to draw a conclusion. The technical and developmental traditions, however, are richly explored through conversations with the interview subjects. The lack of industry standards has meant that these teams are required to use their existing knowledge and expertise in developing and testing the systems they are designing. To achieve this the testers are relying heavily on the three simulations mentioned above and the interactions between those systems to conduct their work. This means that, again as mentioned above, there is inherently a posthumanist reliance on the machines working with the testers but also with each other. In order to build systems and in order to facilitate the interactions between different simulations, the development team needs to consider not only the device but the process and culture that surround that device, the roots of

[35] This sentiment is expressed more completely in *The Art of Software Testing* by Glen Myers, Corey Sandler, and Tom Badgett. On the psychology of testing they say "When you test a program, you want to add some value to it. Adding value through testing means raising the quality or reliability of the program. Raising the reliability of the program means finding and removing errors -**Testing is the process of executing a program with the intent of finding errors**". They go on though to make, to a greater or lesser extent, the argument I've made here – that achieving that goal in an environment of increasing complexity with briefer times to delivery makes the testing challenge one which is inevitably going to cause system failure unless novel testing mechanisms are introduced.

these traditions in the system's development, and how these both concurrently impact the development team's work. [36]

The role of simulation testing is critical, and so I will revisit the TACPS Simulation Deployment diagram introduced in hypothesis #1 (Figure 8), but this time I will frame that discussion from the perspective of hypothesis #3. Considering the three types of simulation testing outlined above we should note that each is drawn from a different tradition. Real world scenario testing is by far the oldest of the three, tracing its roots back to basic principles of the scientific method and the rich tradition (heavily influenced by engineering) of using prototypes and real-world analogs to conduct testing of systems, prototypes, and other goals.[37] Virtual environment simulation testing is rooted in the need to generate results quickly, something that cannot be done through normal iterative prototyping. Where real world-testing (which is analogous to traditional prototyping) is utilized today primarily to reproduce results that would be difficult to model virtually (such as comfort of riders, fear, etc.), a virtual model emphasizes the physical performance of the system. According to one company which generates virtual models:

[36] This bleeds into the conversation around ethics in autonomous vehicles and emerging technologies which is not a topic that is of direct concern to this work. However, it is important to note here the deep body of work that has considered this question, and the intersection of human decisions in machine and technology design that can inform the biased outcome of the technology. Footnote 15 discusses this bias in detail. Here I will only add a few citations for works that discuss the evolution of ethical problems in this space (Brey, 2017; Himmelreich, 2018; Li, J., Zhao, Cho, Ju, & Malle, 2016)

[37] There are too many examples of real-world testing that I could point to here to illustrate the rich tradition from which this type of testing has evolved. However, since much of the discussion has been focused on the autonomotive industry I think a useful example would be the history of the crash test dummy. First created by the US Airforce in 1949 to test the evaluation of aircraft ejection seats on rocket sled tests, and named Sierra Sam, crash test dummies have since served as analogs for humans in real world crash testing. As crash testing technologies continue to be equipped with more and more sensors, the convergence with digital twin technologies as discussed above becomes more likely (Lawton, 2018)

> "This means you know in the design phase whether or not your product will achieve the desired performance. In this phase of the process, you can change one or more parameters and repeat the virtual testing until the performance is all right. The cost of modifying the design in this phase is a fraction of the cost of redesigning a product in the prototype phase. Thanks to the virtual model, we know exactly which design parameters affect the performance and which modifications are most effective to optimize the product performance." (Reden virtual testing.2020)

Virtual testing allows for rapid prototyping and grew out of the tradition that was focused on prioritizing speed. Real-world testing addresses the collection of data that might otherwise not be modeled. Neural networks, by far the most recent of the three, born mostly from AI traditions, and discussed in the literature review above, contribute the important characteristic of being able to recognize correlations and hidden patterns in raw data that might be missed by human testers, and they also are able to create clusters and classifications of data that help to continuously improve their performance. They can be seen as automating the testing process, or at least are a key structure in enabling this automation.

For all three simulation testing types to be employed simultaneously requires that they themselves pass a subsumption test – meaning that each type of simulation test cannot function as a subtype of another test (or that they cannot be redundant in their role in the testing ecosystem) (Ammann & Offutt, 2016). Drawing on three different developmental traditions and seeing how each contributes a unique component of the testing environment, it is clear to see that this triad of testing passes the subsumption test. Autonomous cyber-physical systems require taking the best components of other testing cultures and combining them in a way that allows for a more comprehensive testing. This is a core tenant shared by all those who were interviewed and is reflected in the methods outlined throughout this work. Further, through these interviews we have gained insight into how these three types of testing interplay with one another, and what those different interplays entail.

4.5 - Summarized Significant Findings

This work captures a snapshot in the evolution of the testing of autonomous cyber-physical systems. These technologies, still in their early days of development and deployment, are in the midst of reconciling a diverse array of technological traditions and approaches to quickly and safely solve some monumental engineering challenges. The significant findings from this work begin with the recognition that the control software for autonomous cyber-physical systems have features that are significantly different from traditional application programs. These include the embedding of machine learning components in the form of artificial neural networks, the need to process sensor-based, streaming input data, and the need to deal with a huge number of input variables based on the autonomous CPS characteristics, the environmental characteristics, and the speed at which the autonomous CPS must make accurate decisions. It is clear that with the safety concerns involved and the complexity of the decision making, a multi-layer software design that includes a monitoring layer is necessary to prevent potentially tragic accidents.

A second critical finding from this work is that the role of simulations, in the most advanced applications, works seamlessly to deliver comprehensive testing environments designed to overcome the limitations of time and the gordian knot of interlaced systems and system requirements that are necessary to produce the . With all of these unusual features, it would be impossible to test and evaluate the autonomous CPS's control software with only traditional software testing techniques. Indeed, it raises the question of what software testing and evaluation means when the software includes embedded ANNs being used for machine learning? Must we now consider the machine learning process itself to be part of software testing and evaluation? However one wants to define software testing and evaluation in this software environment, it is clear beyond traditional software testing techniques, the main feature of software testing and evaluation in this new type of testing environment must be approached with multi-dimensional simulation. It is impractical to conduct TACPS simply by operating the system. Not only must multiple types of simulation be employed to comprehensively test the depth of these systems (such as by achieving a fully digital twin) but the simulations must include an expansive set of input variables and input variable values. Even more challenging, the tests of the system including all the permutations must be run a number of times that far exceeds the requirements of traditional software testing and evaluation.

Additionally, this increased complexity and the addition of machine communication with humans out of the loop represents an opportunity for the testing industry to evaluate posthumanist concerns about the role of the machine in the testing process, and whether we need to begin to consider not only the way that these systems interact with us, but the way that they interact with each other. This means, at the risk of adding an additional layer of complexity to the already complicated mesh, that we need to consider the evolution of the approaches that we take in TACPS and that we consider the human as a decentralized part of the experience. As humans are no longer the end users for many of these autonomous systems, a human-centric approach seems too narrow an approach for testing.

It is clear that other aspects of what we normally think of as software testing and evaluation must be added to or expanded. With the greatly increased richness of input variables, such as different terrains, lighting conditions, and weather conditions, exploratory testing and the mutation of input variable values must be taken to a new, higher level of consideration and execution. Another issue, returning to the neural networks feature of the software, is how we define “coverage.” Coverage can no longer mean only branch and path coverage through program code but must also now mean coverage of the neurons in a neural network.

4.6- Recommended Best Practices

To help with the implementation of testing autonomous cyber-physical systems, a few best practices for the testing community have been distilled here. The identified best practices of software testing and evaluation in this space are a natural outgrowth of the significant findings described above. First, it is important to realize that there is a continuum of simulation, that simulation is a necessary component for TACPS, and that it is not sufficient to conduct a single type of simulation in the development of complex systems. Carefully constructed simulations of TACPS, their environments, and the streaming data input are required. Also required is the recognition that the output of a particular set of input values may be indeterminate with multiple acceptable results. Traditional testing approaches can be used to a limited extent, but the layered complexity of the system requires the introduction of automation, especially at scale. Test case generation, variable testing, and other mechanisms are exponentially more complicated for these systems, and because of the tremendous safety requirements placed on many of these systems, the testing must be thorough and exact. Compressing the volume of required testing into time parameters that are realistic for industry requires not only multiple simulation types, but automation in the evolution of the interactions of these systems so that the human tester is mostly out of the loop.

As the tester is moved outside of the loop, it is also important that we consider more deeply the evolution of the interaction between the system tester and the system itself. Understanding how humans interact with the autonomous systems, and considering how the systems interact with each other, will afford more clarity in constructing the best test environment.

4.7 - Identified Challenges and Future Research

The single biggest challenge in conducting this study was securing the cooperation of a diverse pool of participants that was of sufficient size. Especially among corporate and industry-based testers, we found that there were tremendous hurdles in getting access to willing participants. Our team contacted at least 40 companies and other organizations to request participation and only a handful responded, and of those only the seven responses came back. In one case, a person in industry who we interviewed stopped at one point because he had previously worked for the government and he was concerned that he might divulge classified information.

Another challenge that we found in conducting the interviews was that in many cases there was a difference in vocabulary around testing which could stem in part from the diversity of experiences and traditions required for TACPS. As a result, some questions in the survey were difficult to parse for testers, which slowed or stopped the interview process in certain cases. In other cases, the questions (such as some questions provided by AFIT as part of the survey) did not use language that testers were familiar with. This is in itself an interesting difference that warrants further lexographical and etymological exploration. A third challenge was that the exploration of posthumanism in the testing culture was not sufficiently probed through the questions that were asked. Though there did appear to be rich data to explore here, it was not a major part of the results or analysis. Finally, the subject of exploratory testing was covered, but there was insufficient data in the surveys to warrant a sufficient exploration.

As far as future studies are concerned, at many points throughout this work I have indicated areas where there might be potential for deeper exploration. The novelty of applying ANN in testing warrants further exploration – we are in the midst of an evolution and being able to measure and write about that is rare for any researcher. Understanding not only how these techniques are being implemented into testing architectures, but also learning more about the modifications and specific instances of the technology would be incredibly valuable. Additionally, understanding and reconciling the three types of simulation outlined here, and more importantly understanding the interaction between the three and how these interactions manifest in new techniques like digital twins is a new frontier for testing research that is promising. The posthumanist perspective introduced deserves a more detailed exploration and consideration in the context of the increasingly independent and interconnected role of autonomous cyber-physical systems. More specifically, considering the posthumanist paradigm in TACPS will lead to greater exploration of the increasing interplay between humans and machines as equal partners in building testing architecture. Finally, at the outset of the study the role of exploratory testing was thought to be a richer source of discussion than what the interviews provided, and this is likely because of the small sample size. Further exploration of this topic is required.

Section 5 - Conclusion

I have attempted in this work to chart the evolution of the testing of autonomous cyber-physical systems and through interviews with industry professionals to explore the current best practices of industry. What we have found in this study is a complex interaction between multiple types of simulation testing that represents the cutting edge of industrial practice. Autonomous cyber-physical systems include the most complex testing environments being explored today. The pioneers building these testing environments are marrying different developmental traditions and approaches with automation to overcome otherwise insurmountable development timelines placed on technologies that are not fully understood, that represent a potential danger with still unclear safety requirements, and that if deployed correctly could transform the world in which we live. The stakes are high, and so understanding and adopting best practices from industry practitioners at the forefront of systems testing helps us to better plan for the future of TACPS.   This study has shown that there is unique interconnection between the three types of simulation testing that enables advanced techniques. It has also identified several areas of further research which will help to more fully outline the trajectory of this emerging field.

References

Ackoff, R. L. (1967a). Management misinformation systems. *Management Science, 14*(4), B-156.

Ackoff, R. L. (1967b). Management misinformation systems. *Management Science, 14*(4), B-156.

Ammann, P., & Offutt, J. (2016). *Introduction to software testing* Cambridge University Press.

Aniculaesei, A., Grieser, J., Rausch, A., Rehfeldt, K., & Warnecke, T. (2018). (2018). Toward a holistic software systems engineering approach for dependable autonomous systems. Paper presented at the *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS),* 23-30.

Apuzzo, J. T., Marino, J. P., Hoskins, C. L., Race, T. L., & Suri, H. R. (2005). No title. *Method and Apparatus for Testing a Software Component using an Abstraction Matrix,*

Asimov, I. (2004). *I, robot* Spectra.

Attia, M., Hossny, M., Nahavandi, S., Dalvand, M., & Asadi, H. (2018). (2018). Towards trusted autonomous surgical robots. Paper presented at the *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC),* 4083-4088.

Austrin, T., & Farnsworth, J. (2005). Hybrid genres: Fieldwork, detection and the method of bruno latour. *Qualitative Research, 5*(2), 147-165.

Bach, J. (2003). No title. *Exploratory Testing Explained,*

Bagheri, B., Yang, S., Kao, H., & Lee, J. (2015). Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. *IFAC-PapersOnLine, 48*(3), 1622-1627.

Baheti, R., & Gill, H. (2011). Cyber-physical systems. *The Impact of Control Technology, 12*(1), 161-166.

Bajracharya, M., Maimone, M. W., & Helmick, D. (2008). Autonomy for mars rovers: Past, present, and future. *Computer, 41*(12), 44-50.

Bandyopadhyay, P. R. (2005). Trends in biorobotic autonomous undersea vehicles. *IEEE Journal of Oceanic Engineering, 30*(1), 109-139.

Bazeley, P., & Jackson, K. (2013). *Qualitative data analysis with NVivo* SAGE publications limited.

Beck, K. (2003). *Test-driven development: By example* Addison-Wesley Professional.

Beydeda, S., & Gruhn, V. (2001). (2001). An integrated testing technique for component-based software. Paper presented at the *Proceedings ACS/IEEE International Conference on Computer Systems and Applications,* 328-334.

Boix Mansilla, V., Lamont, M., & Sato, K. (2016). Shared cognitive–emotional–interactional platforms: Markers and conditions for successful interdisciplinary collaborations. *Science, Technology, & Human Values, 41*(4), 571-612.

Boland Jr, R. J. (1978). The process and product of system design. *Management Science, 24*(9), 887-898.

Boysen, N., Schwerdfeger, S., & Weidinger, F. (2018). Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research, 271*(3), 1085-1099.

Brey, P. (2017). Ethics of emerging technology. *The Ethics of Technology: Methods and Approaches,* , 175-191.

Brotton, J. (2013). *A history of the world in 12 maps* Penguin.

Brunner, G., Szebedy, B., Tanner, S., & Wattenhofer, R. (2019). (2019). The urban last mile problem: Autonomous drone delivery to your balcony. Paper presented at the *2019 International Conference on Unmanned Aircraft Systems (ICUAS),* 1005-1012.

Burnstein, I. (2006). *Practical software testing: A process-oriented approach* Springer Science & Business Media.

Canfora, G., Mercaldo, F., Visaggio, C. A., DAngelo, M., Furno, A., & Manganelli, C. (2013). (2013). A case study of automating user experience-oriented performance testing on smartphones. Paper presented at the *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation,* 66-69.

Card, S. K. (2018). *The psychology of human-computer interaction* Crc Press.

Chander, A. (2016). The racist algorithm. *Mich.L.Rev., 115*, 1023.

Charmaz, K. (2009). Shifting the grounds: Constructivist grounded theory methods. *Developing Grounded Theory: The Second Generation,* , 127-154.

Chatterjee, R., Arun, G., Agarwal, S., Speckhard, B., & Vasudevan, R. (2004). (2004). Using data versioning in database application development. Paper presented at the *Proceedings. 26th International Conference on Software Engineering,* 315-325.

Chesbrough, H. (2017). The future of open innovation: The future of open innovation is more extensive, more collaborative, and more engaged with a wider variety of participants. *Research-Technology Management, 60*(1), 35-38.

Chien, T., Wang, H., Chang, Y., & Kan, W. (2019). Using google maps to display the pattern of coauthor collaborations on the topic of schizophrenia: A systematic review between 1937 and 2017. *Schizophrenia Research, 204*, 206-213.

Crispin, L., & Gregory, J. (2009). *Agile testing: A practical guide for testers and agile teams* Pearson Education.

Cui, B., Yokoi, S., & Kato, J. (2009). (2009). Integrating correlative knowledge in virtual museum with google map. Paper presented at the *Proceedings of Annual Conference of Japan Association for Social Informatics Proceedings of the 24th Annual Conference of Japan Association for Social,* 48-53.

de Azevedo, M. S. (2010). *Men of a single book: Fundamentalism in islam, christianity, and modern thought* World Wisdom, Inc.

de Vasconcelos Gomes, Leonardo Augusto, Facin, A. L. F., Salerno, M. S., & Ikenami, R. K. (2018). Unpacking the innovation ecosystem construct: Evolution, gaps and trends. *Technological Forecasting and Social Change, 136*, 30-48.

Decker, M. (2008). Caregiving robots and ethical reflection: The perspective of interdisciplinary technology assessment. *AI & Society, 22*(3), 315-330.

Decker, M., & Grunwald, A. (2001). Rational technology assessment as interdisciplinary research. *Interdisciplinarity in technology assessment* (pp. 33-60) Springer.

Dhaliwal, J., Onita, C. G., Poston, R., & Zhang, X. P. (2011). Alignment within the software development unit: Assessing structural and relational dimensions between developers and testers. *The Journal of Strategic Information Systems, 20*(4), 323-342.

Dixon, L. (2020). Autonowashing: The greenwashing of vehicle automation. *Transportation Research Interdisciplinary Perspectives, 5*, 100113.

Drechsler, A., Hevner, A. R., & Gill, T. G. (2016). (2016). Beyond rigor and relevance: Exploring artifact resonance. Paper presented at the *2016 49th Hawaii International Conference on System Sciences (HICSS),* 4434-4443.

Drucker, P. F. (1988). The coming of the new organization.

Durst, P. J., & Gray, W. (2014). No title. *Levels of Autonomy and Autonomous System Performance Assessment for Intelligent Unmanned Systems,* Felix, C., Franconeri, S., & Bertini, E. (2017). Taking word clouds apart: An empirical investigation of the design space for keyword summaries. *IEEE Transactions on Visualization and Computer Graphics, 24*(1), 657-666.

Ferrando, F. (2012). Towards a posthumanist methodology. A statement. *Frame Journal for Literary Studies, 25*(1), 9-18.

French, R. M. (1990). Subcognition and the limits of the turing test. *Mind, 99*(393), 53-65.

Galbraith, J. R. (1974). Organization design: An information processing view. *Interfaces, 4*(3), 28-36.

Gao, J., Tsao, H., & Wu, Y. (2003). *Testing and quality assurance for component-based software* Artech House.

Gilb, T., & Finzi, S. (1988). *Principles of software engineering management* Addison-wesley Reading, MA.

Gillenson, M. L., Racer, M. J., Zhang, X., Booth, R. E., & Dugan, J. P. (2016). A heuristic method for scheduling requirements implementation in agile software development projects. *Journal of Information Technology Management, 27*(4), 169.

Gillenson, M. L., Stafford, T. F., & Shi, Y. (2020). *Use of qualitative research to generate a function for finding the unit cost of software test cases* IGI Global.

Glaser, B. G., & Strauss, A. L. (2017). *Discovery of grounded theory: Strategies for qualitative research* Routledge.

Gray, D. E. (2019). *Doing research in the business world* Sage Publications Limited.

Grudin, J. (2008). A moving target: The evolution of HCI. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications,* , 1-24.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645-1660.

Hadjigeorgiou, J., & Potvin, Y. (2007). (2007). Overview of dynamic testing of ground support. Paper presented at the *Proceedings of the Fourth International Seminar on Deep and High Stress Mining,* 349-371.

Haemer, K. W. (1949). Presentation problems: Area bias in map presentation. *The American Statistician, 3*(2), 19.

Hai-Jew, S. (2017). Employing the sentiment analysis tool in nvivo 11 plus on social media data: Eight initial case types. *Social media listening and monitoring for business applications* (pp. 175-244) IGI Global.

Hameed, M. A., Counsell, S., & Swift, S. (2012). A conceptual model for the process of IT innovation adoption in organizations. *Journal of Engineering and Technology Management, 29*(3), 358-390.

Harrington, H. J. (1994). *Business process improvement* Association for Quality and Participation.

Harris, J. (2011). Word clouds considered harmful. *Nieman Journalism Lab, 13*

Hass, A. M. (2014). *Guide to advanced software testing* Artech House.

Hayes, P., & Ford, K. (1995). (1995). Turing test considered harmful. Paper presented at the *Ijcai (1),* 972-977.

Hayles, N. K. (2008). *How we became posthuman: Virtual bodies in cybernetics, literature, and informatics* University of Chicago Press.

Heath, H., & Cowley, S. (2004). Developing a grounded theory approach: A comparison of glaser and strauss. *International Journal of Nursing Studies, 41*(2), 141-150.

Hernandez-Orallo, J. (2000). Beyond the turing test. *Journal of Logic, Language and Information, 9*(4), 447-466.

Himmelreich, J. (2018). Never mind the trolley: The ethics of autonomous vehicles in mundane situations. *Ethical Theory and Moral Practice, 21*(3), 669-684.

Hoffer, J. A., Ramesh, V., & Topi, H. (2019). *Modern database management 13th ed.* Upper Saddle River, NJ: Prentice Hall,.

Hummel, O., Eichelberger, H., Giloj, A., Werle, D., & Schmid, K. (2018). (2018). A collection of software engineering challenges for big data system development. Paper presented at the *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA),* 362-369.

Hutchison, C., Zizyte, M., Lanigan, P. E., Guttendorf, D., Wagner, M., Le Goues, C., & Koopman, P. (2018). (2018). Robustness testing of autonomy software. Paper presented at the *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP),* 276-285.

Iliadis, A. (2018). Algorithms, ontology, and social progress. *Global Media and Communication, 14*(2), 219-230.

Jamshed, S. (2014). Qualitative research method-interviewing and observation. *Journal of Basic and Clinical Pharmacy, 5*(4), 87.

Johnson, B., & Gray, R. (2010). A history of philosophical and theoretical issues for mixed methods research. *Sage Handbook of Mixed Methods in Social and Behavioral Research, 2*, 69-94.

Kaner, C. (2008). A tutorial in exploratory testing. *Tutorial Presented at QUEST2008.(Available Online at: Http://Www.Kaner.Com/Pdfs/QAIExploring.Pdf, Accessed: 26 Jan 2014),*

Kaner, C., Falk, J., & Nguyen, H. Q. (2000). *Testing computer software second edition* Dreamtech Press.

Kaplan, B., & Duchon, D. (1988). Combining qualitative and quantitative methods in information systems research: A case study. *MIS Quarterly,* , 571-586.

Kast, F. E., & Rosenzweig, J. E. (1972). General systems theory: Applications for organization and management. *Academy of Management Journal, 15*(4), 447-465.

Kirk, G. S. (1951). Natural change in heraclitus. *Mind, 60*(237), 35-42.

Koopman, P., Ferrell, U., Fratrik, F., & Wagner, M. (2019). (2019). A safety standard approach for fully autonomous vehicles. Paper presented at the *International Conference on Computer Safety, Reliability, and Security,* 326-332.

Koopman, P., & Wagner, M. (2016). Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety, 4*(1), 15-24.

Krasniqi, X., & Hajrizi, E. (2016). Use of IoT technology to drive the automotive industry from connected to full autonomous vehicles. *IFAC-PapersOnLine, 49*(29), 269-274.

Kundu, S., Mak, T. M., & Galivanche, R. (2004). (2004). Trends in manufacturing test methods and their implications. Paper presented at the *2004 International Conferce on Test,* 679-687.

Latour, B. (2000). The berlin key or how to do words with things. *Matter, Materiality and Modern Culture,* , 10-21.

Lawton, G. (2018). Crash test dummies. *New Scientist, 238*(3177), 42-43.

Lee, A. S. (1989). A scientific methodology for MIS case studies. *MIS Quarterly,* , 33-50.

Lee, J., Bagheri, B., & Kao, H. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters, 3*, 18-23.

Leech, B. L. (2002). Asking questions: Techniques for semistructured interviews. *PS: Political Science & Politics, 35*(4), 665-668.

Lewis, W. E. (2017). *Software testing and continuous quality improvement* Auerbach publications.

Li, J., Zhao, X., Cho, M., Ju, W., & Malle, B. F. (2016). No title. *From Trolley to Autonomous Vehicle: Perceptions of Responsibility and Moral Norms in Traffic Accidents with Self-Driving Cars,*

Li, Z., Ma, X., Xu, C., Cao, C., Xu, J., & Lü, J. (2019). (2019). Boosting operational dnn testing efficiency through conditioning. Paper presented at the *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering,* 499-509.

Lim, L. Y. (1968). A pathfinding algorithm for a myopic robot.

Lopian, E. (2013). No title. *Method and System for Isolating Software Components,*

Lucas, H. C., Ginzberg, M., & Schultz, R. (1991). Implementing information systems: Testing a structural model. *Norwood, NJ: Ablex,*

Ma, L., Juefei-Xu, F., Zhang, F., Sun, J., Xue, M., Li, B., . . . Liu, Y. (2018). (2018). Deepgauge: Multi-granularity testing criteria for deep learning systems. Paper presented at the *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering,* 120-131.

Malhotra, A., Gosain, S., & Sawy, O. A. E. (2005). Absorptive capacity configurations in supply chains: Gearing for partner-enabled market knowledge creation. *MIS Quarterly,* , 145-187.

Matook, S., & Brown, S. A. (2008). Conceptualizing the IT artifact for MIS research. *ICIS 2008 Proceedings,* , 102.

Mazzei, L. A. (2013). A voice without organs: Interviewing in posthumanist research. *International Journal of Qualitative Studies in Education, 26*(6), 732-740.

McKay, J., Marshall, P., & Hirschheim, R. (2012). The design construct in information systems design science. *Journal of Information Technology, 27*(2), 125-139.

Meyer, A. (2019). PILGRIMAGE AND ARCHAEOLOGY-(TM) kristensen,(W.) friese (edd.) excavating pilgrimage. archaeological approaches to sacred travel and movement in the ancient world. pp. xiv 291, ills, maps. london and new york: Routledge, 2017. cased,£ 120, US $150. ISBN: 978-1-4724-5390-7. *The Classical Review, 69*(2), 581-584.

Miller, C. C. (2006). A beast in the field: The google maps mashup as GIS/2. *Cartographica: The International Journal for Geographic Information and Geovisualization, 41*(3), 187-199.

Moghaddam, A. (2006). Coding issues in grounded theory. *Issues in Educational Research, 16*(1), 52-66.

Myers, M. D. (1997a). Critical ethnography in information systems. *Information systems and qualitative research* (pp. 276-300) Springer.

Myers, M. D. (1997b). Qualitative research in information systems. *Management Information Systems Quarterly, 21*(2), 241-242.

Naveen. (2015). What is V model in software testing and what are advantages and disadvantages of V model. Retrieved from http://testingfreak.com/v-model-software-testing-advantages-disadvantages-v-model/

Nidhra, S., & Dondeti, J. (2012). Black box and white box testing techniques-a literature review. *International Journal of Embedded Systems and Applications (IJESA), 2*(2), 29-50.

Nolan, R. L. (1979). Managing the crises in data processing. *Managing the Crises in Data Processing.Harvard Business Review, March-April 1979, Pp.115-126,*

Nordstrom, S. N. (2015). A data assemblage. *International Review of Qualitative Research, 8*(2), 166-193.

Norvig, P. R., & Intelligence, S. A. (2002). *A modern approach* Prentice Hall.

Nystrom, D., Nolin, M., Tesanovic, A., Norstrom, C., & Hansson, J.Pessimistic concurrency control and versioning to support database pointers in real-time databases. Paper presented at the *Proceedings. 16th Euromicro Conference on Real-Time Systems, 2004. ECRTS 2004.* 261-270.

O'Lear, S. R. (1996). Using electronic mail (e-mail) surveys for geographic research: Lessons from a survey of russian environmentalists. *The Professional Geographer, 48*(2), 209-217.

Oyebode, O., Patrick, H., Walker, A., Campbell, B., & Powell, J. (2016). The ghost in the machine? the value of expert advice in the production of evidence-based guidance: A mixed methods study of the NICE interventional procedures programme. *International Journal of Technology Assessment in Health Care, 32*(1-2), 61-68.

Paltoglou, G., & Thelwall, M. (2010). (2010). A study of information retrieval weighting schemes for sentiment analysis. Paper presented at the *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* 1386-1395.

Palvia, P., Kakhki, M. D., Ghoshal, T., Uppala, V., & Wang, W. (2015). Methodological and topic trends in information systems research: A meta-analysis of IS journals. *Cais, 37*, 30.

Pei, K., Cao, Y., Yang, J., & Jana, S. (2017). (2017). Deepxplore: Automated whitebox testing of deep learning systems. Paper presented at the *Proceedings of the 26th Symposium on Operating Systems Principles,* 1-18.

Petty, J. (2017). Getting the best out of autonomous mining fleets. *AusIMM Bulletin,* (Dec 2017), 58.

Posner, R. A. (2000). Florida 2000: A legal and statistical analysis of the election deadlock and the ensuing litigation. *The Supreme Court Review, 2000*, 1-60.

Pratchett, T. (2011). *I shall wear midnight* Random House.

QSR International Pty Ltd. (2018). NVivo qualitative data analysis software [computer software]

Reden virtual testing. (2020). Retrieved from https://www.reden.nl/en/virtueel-testen

Richards, L. (1999). *Using NVivo in qualitative research* Sage.

Royce, W. W. (1987). (1987). Managing the development of large software systems: Concepts and techniques. Paper presented at the *Proceedings of the 9th International Conference on Software Engineering,* 328-338.

Ryan, H. W. (1999). Managing development in the era of large complex systems. *IS Management, 16*(2), 89-91.

Salmons, J. (2009). *Online interviews in real time* Sage.

Salmons, J. (2014). *Qualitative online interviews: Strategies, design, and skills* Sage Publications.

Schön, D. A. (1967). *Technology and change: The new heraclitus* Delta.

Shah, H. (2011). (2011). Turing's misunderstood imitation game and IBM's watson success. Paper presented at the *Keynote in 2nd Towards a Comprehensive Intelligence Test (TCIT) Symposium at AISB,*

Shen, Z., & Zhai, X. (2017). Effectiveness analysis of ground simulation space environmental tests and their effects on spacecraft. *Protection of materials and structures from the space environment* (pp. 489-499) Springer.

Smithers, T. (1997). Autonomy in robots and other agents. *Brain and Cognition, 34*(1), 88-106.

Soneson, G. L., Horn, J. F., & Zheng, A. (2016). Simulation testing of advanced response types for ship-based rotorcraft. *Journal of the American Helicopter Society, 61*(3), 1-13.

Stumpf, S., Burnett, M., Pipek, V., & Wong, W. (2012). End-user interactions with intelligent and autonomous systems. *CHI'12 extended abstracts on human factors in computing systems* (pp. 2755-2758)

Swade, D., & Babbage, C. (2001). *Difference engine: Charles babbage and the quest to build the first computer* Viking Penguin.

Switkes, J. P., Gerdes, J. C., & Berdichevsky, E. (2019). No title. *Methods and Systems for Semi-Autonomous Vehicular Convoys,*

Tabani, H., Kosmidis, L., Abella, J., Cazorla, F. J., & Bernat, G. (2019). (2019). Assessing the adherence of an industrial autonomous driving framework to iso 26262 software guidelines. Paper presented at the *2019 56th ACM/IEEE Design Automation Conference (DAC),* 1-6.

Tai, K., & Lei, Y. (2002). A test generation strategy for pairwise testing. *IEEE Transactions on Software Engineering, 28*(1), 109-111.

Thompson, M. (2008). No title. *Testing the Intelligence of Unmanned Autonomous Systems,*

Tian, Y., Pei, K., Jana, S., & Ray, B. (2018a). (2018a). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. Paper presented at the *Proceedings of the 40th International Conference on Software Engineering,* 303-314.

Tian, Y., Pei, K., Jana, S., & Ray, B. (2018b). (2018b). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. Paper presented at the *Proceedings of the 40th International Conference on Software Engineering,* 303-314.

Trantopoulos, K., von Krogh, G., Wallin, M. W., & Woerter, M. (2017). External knowledge and information technology: Implications for process innovation performance. *MIS Quarterly, 41*(1), 287-300.

Trauth, E. M. (2001). The choice of qualitative methods in IS research. *Qualitative research in IS: Issues and trends* (pp. 1-19) IGI Global.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*(236), 433-460. Retrieved from https://doi.org/10.1093/mind/LIX.236.433

Twain, M. (1876). *The adventures of tom sawyer*
(2007th ed.) OUP Oxford.

Valdez, A. C., Schaar, A. K., Ziefle, M., & Holzinger, A. (2014). (2014). Enhancing interdisciplinary cooperation by social platforms. Paper presented at the *International Conference on Human Interface and the Management of Information,* 298-309.

Wang, F., Zhang, J. J., Zheng, X., Wang, X., Yuan, Y., Dai, X., . . . Yang, L. (2016). Where does AlphaGo go: From church-turing thesis to AlphaGo thesis and beyond. *IEEE/CAA Journal of Automatica Sinica, 3*(2), 113-120.

Weigold, M. F. (2001). Communicating science: A review of the literature. *Science Communication, 23*(2), 164-193.

Wolfe, C. (2010). *What is posthumanism?* U of Minnesota Press.

Xu, L. D., Xu, E. L., & Li, L. (2018). Industry 4.0: State of the art and future trends. *International Journal of Production Research, 56*(8), 2941-2962.

Yampolskiy, R. V. (2016). (2016). Taxonomy of pathways to dangerous artificial intelligence. Paper presented at the *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence,*

Zahra, S. A., & George, G. (2002). Absorptive capacity: A review, reconceptualization, and extension. *Academy of Management Review, 27*(2), 185-203.

Zakaria, F. (2015). *In defense of a liberal education* WW Norton & Company.

Zeng, A., Song, S., Yu, K., Donlon, E., Hogan, F. R., Bauza, M., . . . Romo, E. (2018). (2018). Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. Paper presented at the *2018 IEEE International Conference on Robotics and Automation (ICRA),* 1-8.

Appendix A – Survey Questions

Questionnaire for the Industrial Study

University of Memphis AFIT Study Questionnaire

Study of the Testing of Software in Autonomous Cyber-Physical Systems

As part of a task for the Air Force Institute of Technology Scientific Test and Analysis Techniques (STAT) Center of Excellence, the University of Memphis is conducting a survey of best practices in government and industry regarding software testing of systems with autonomy. It would be most helpful to our effort to learn your answers to the following questions.

Interview Form

1. Your Autonomous Cyber-Physical System (ACPS).
    a. Please provide a brief overview of your ACPS, including how it is expected to relate to its environment.
    b. At what stage of development is your ACPS?
    c. Is there a non-autonomous version of your ACPS?
    d. What types of sensors are you using on your ACPS that provide streaming input data?

2. Your Input Data.
    a. Describe the streaming input data being fed into your software.
    b. Describe any other input data being fed into your software.

3. Your Software.
    a. How does your software process streaming data?
    b. Does your software include elements of machine learning or other aspects of artificial intelligence?
    c. Is your software multi-layered, such as with one layer used to control the ACPS and one or more other layers used to monitor the first layer to make sure that it doesn't do anything dangerous?
    d. How do you understand a non-autonomous version of your system from a software perspective and does that understanding inform your approach to the autonomous system?

4. Your Software Testing Methodologies.
    a. Describe your approach to testing the software in ACPS.
    b. Are you using simulations to do your testing? If so, what are you using and how are you using them?

c. Are you using machine learning or artificial intelligence in performing software testing? Please explain.
d. Are you employing exploratory testing and, if so, how? (Please clarify your definition of "exploratory testing".)
e. Are you employing any other specialized software testing methods? Please explain.

5. Your Software Testing Execution.
   a. What is the nature of your test cases and how do you create them?
   b. Do you attempt to isolate individual test cases in the environment of streaming sensor data?
   c. What is your methodology regarding the expected outcomes of test cases, i.e. test oracles?
   d. How is your test data reduced and analyzed and what the outputs of that analysis?
   e. How do you determine success or failure of your system based on test results?
   f. How do you ensure adequate coverage of your applications with test cases?

6. Software Testing Standards and Measurements.
   a. What standards, if any, do you use in testing your software or system?
   b. How do you determine the effectiveness of the software testing methods you are using?
   c. How do you quantify which software testing methods are better than others?
   d. In addition to testing functional requirements (Measures of Effectiveness (MOE)), do you also test performance (Measures of Performance (MOP)), security, or any other non-functional requirements? Please explain.

7. General, Open-Ended Question.
   a. Provide some examples of challenges or gaps in methods, processes or infrastructure you have encountered in T&E of autonomous cyber-physical systems, and how those challenges or gaps are being addressed.

Addendum with Specific Choices:

Engineering approach of your ACPS

Please select all that apply

□ Model-based systems engineering (MBSE)
□ Modeling and simulation-based systems engineering (M&SBSE)
□ Document-based information exchange
□ Other ______________

Components of your ACPS

Please select all that apply

□ Hardware interface/component
□ Software interface/component
□ Network interface/component
□ Other ______________

Levels of your ACPS[38]

Please select all that apply

□ Smart connection Level
□ Data-to-Information Conversion Level
□ Cyber Level
□ Cognition Level
□ Configuration Level
□ Other ______________

Attributes of your ACPS

Please select all that apply

□ Self-aware
□ Self-predict
□ Self-compare
□ Self-configure
□ Self-maintain
□ Self-organize
□ Other ______________

Testing methods of your ACPS[39]

Please select all that apply

□ Model-based
□ Search-based
□ Monitor-based
□ Fault injection-based
□ Big data-driven
□ Cloud-based
□ Other ______________

Test techniques used in testing your ACPS

Please select all that apply

□ Equivalence Partitioning
□ Boundary Value Analysis
□ Decision Tables
□ Cause-Effect Graphing

[38] CPS 5C level architecture from A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems

[39] Review on Testing of Cyber-Physical Systems: Methods and Testbeds

□ State Transition Testing
□ Combinatorial Testing Techniques
□ Use Case Testing
□ User Story Testing
□ Domain Analysis
□ Security testing
□ Conformance testing
□ Robustness testing
□ Combining Techniques
□ Other ______________

Test tools used in testing your ACPS

Please select all that apply

□ Commercial software test/simulation tools
□ Open source software test/simulation tools
□ Project-built software test/simulation tools
□ Other ______________

Test bed used in testing your ACPS

Please select all that apply

□ Security-Oriented Testbed
□ Control-Oriented Testbed
□ Performance-Oriented Testbed
□ Multi-Objective Comprehensive Testbed
□ Other ______________

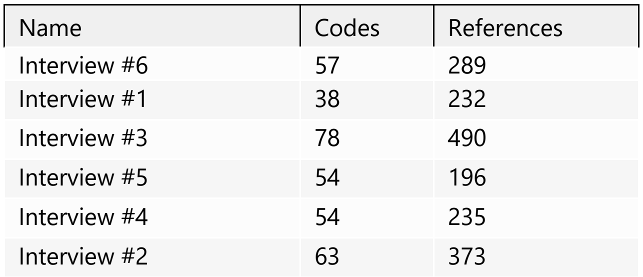
What degree of effectiveness would you assign to your ACPS testing?

Please select the one that best applies

□ Highly effective
□ Effective
□ Moderately effective
□ Not effective

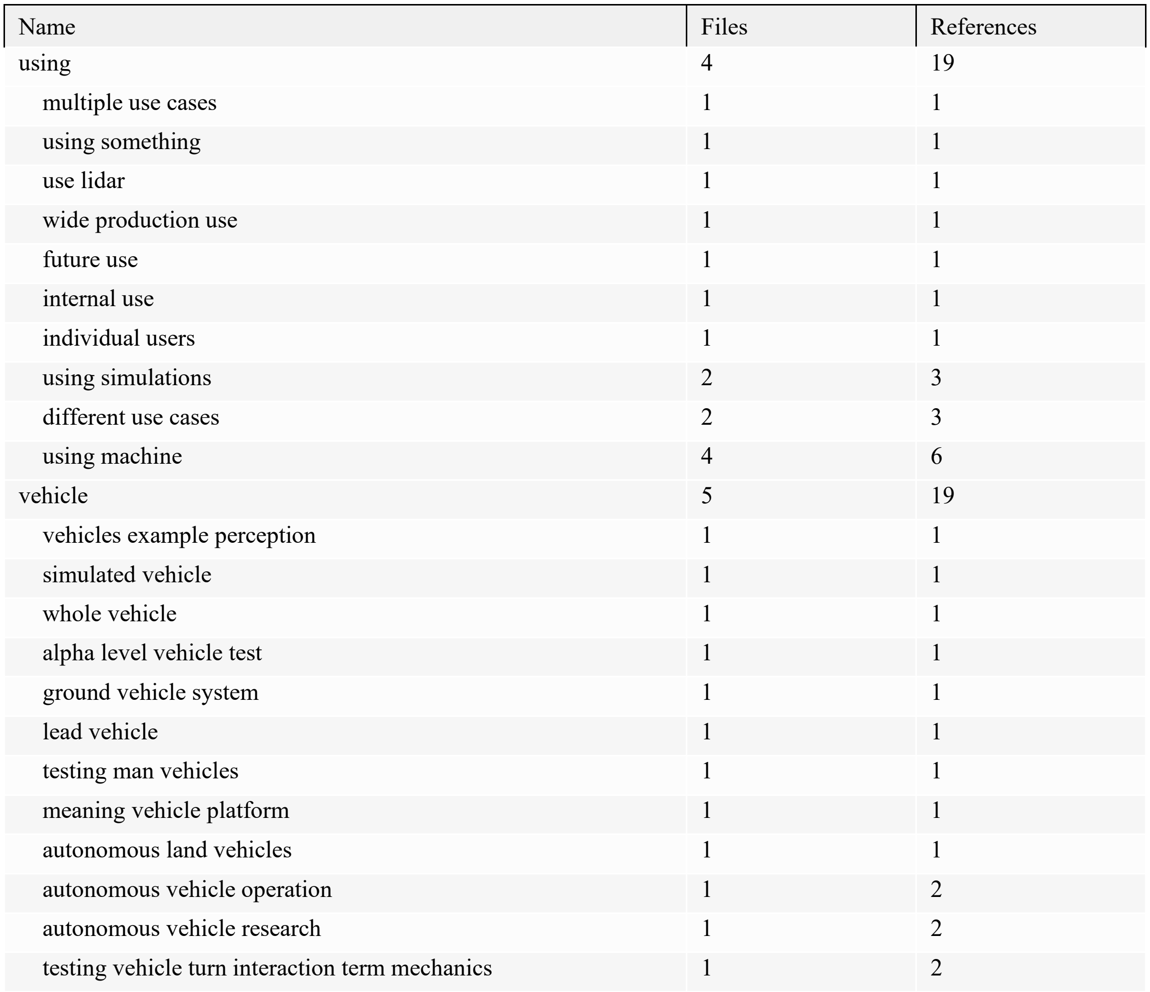
Appendix B

Codes and references counts for each interview



Name	Codes	References
Interview #6	57	289
Interview #1	38	232
Interview #3	78	490
Interview #5	54	196
Interview #4	54	235
Interview #2	63	373

Meta Categories and codebook for each category including reference count



Name	Files	References
using	4	19
multiple use cases	1	1
using something	1	1
use lidar	1	1
wide production use	1	1
future use	1	1
internal use	1	1
individual users	1	1
using simulations	2	3
different use cases	2	3
using machine	4	6
vehicle	5	19
vehicles example perception	1	1
simulated vehicle	1	1
whole vehicle	1	1
alpha level vehicle test	1	1
ground vehicle system	1	1
lead vehicle	1	1
testing man vehicles	1	1
meaning vehicle platform	1	1
autonomous land vehicles	1	1
autonomous vehicle operation	1	2
autonomous vehicle research	1	2
testing vehicle turn interaction term mechanics	1	2

autonomous vehicles	3	4
level	5	22
alpha level vehicle test	1	1
system level design team	1	1
enterprise levels	1	1
customer level	1	1
information conversion diver cognition level configuration	1	1
information conversion level cyber level cognition	1	1
level kind	1	1
connection level	1	1
cyber levels	1	1
cognition level	1	1
increasing levels	1	1
technical level	1	1
levels cognition levels	1	1
configuration level	2	2
information conversion level	2	2
comp configuration level	1	2
smart connection level data	2	3
sensors	6	23
sensor number	1	1
redundant sensors	1	1
actual sensors	1	1
different sensors	1	1
sensor products	1	1
forward sensors	1	1
inertial sensors	1	1
beam sensor	1	1
additional sensor	1	1
camera sensors	1	1
regular sensors	1	1
main sensors	1	1
standard sensors	1	1
sensor types	1	1
full sensor	1	1
multiple sensors	1	1
radar sensors	1	1
sensor hardware	1	2
lidar sensors	1	2
sensor data	2	2
software	6	25
system software	1	1
software interface	1	1

additional software	1	1
software application	1	1
staff software stack	1	1
standard software development tools	1	1
software architecture	1	1
testing computer software	1	1
software checking	1	1
software defects event	1	1
software testing culture	1	1
software failure something	1	1
software response	1	1
software side	1	1
software cover	1	1
software agent	1	1
actually simulation software	1	2
software testing method	1	2
software testing execution	1	2
regarding software testing	2	3
data	6	52
camera data	1	1
clear data	1	1
cloud data	1	1
data collection run	1	1
data collection side	1	1
data flight	1	1
data scrub	1	1
direct data	1	1
discrete data packets	1	1
driver input data	1	1
geometric data	1	1
lidar data	1	1
manufacturer spec data	1	1
map data	1	1
numeric data	1	1
objects data	1	1
physical data	1	1
radar data	1	1
raw data	1	1
raw image data	1	1
re-simulated data	1	1
screaming input data	1	1
selected test data	1	1
several data collection	1	1

training data	1	1
useful actionable data	1	1
video data right	1	1
in-house data sets	1	2
raw data stream	1	2
recording data	1	2
sensor data	2	2
streaming data	2	2
streaming input data	2	2
weather data	1	2
smart connection level data	2	3
test data	2	3
data collection	4	5
system	6	53
system software	1	1
system side	1	1
system level design team	1	1
redundant system	1	1
simulation system	1	1
closed system	1	1
original system	1	1
small system	1	1
value system	1	1
modular delivery system	1	1
rtk system	1	1
road system	1	1
ground vehicle system	1	1
mobility systems branch	1	1
perception system	1	1
stereo camera system	1	1
model-based systems engineering modeling	1	1
lidar system	1	1
safety system	1	1
external monitoring system	1	1
lyme system	1	1
conveyor system	1	1
mechanical system	1	1
manufacturing system	1	1
systems engineering	1	1
model-based systems engineering	2	2
different autonomy systems	1	2
model-based systems engineering approach	1	2
lyme shield system	1	2

test system	1	3
physical systems	3	4
cyber-physical systems	4	6
autonomous systems	4	7
testing	6	93
test site	1	1
stress testing	1	1
alpha level vehicle test	1	1
random emissions tests	1	1
test quality	1	1
in-vehicle tests	1	1
parametrization test	1	1
test requirement	1	1
selected test data	1	1
development testing	1	1
unit testing	1	1
ground truth test	1	1
main test	1	1
traverse test	1	1
model test	1	1
testing perspective	1	1
design test	1	1
testing evaluation command	1	1
test track	1	1
autonomous testing	1	1
contractors test facility	1	1
testing man vehicles	1	1
performance-oriented test bed	1	1
comprehensive test bed	1	1
experimental test	1	1
situation tests	1	1
automated testing	1	1
testing piece	1	1
test matrix	1	1
simulation testing	1	1
testing computer software	1	1
testing procedures	1	1
small-scale testing	1	1
different testing cultures	1	1
software testing culture	1	1
defect test	1	1
much testing	1	1
conventional testing	1	1

code coverage test	1	1
testing tools	1	1
purely testing	1	1
traditional test case yeah	1	2
slalom test	1	2
key performance test	1	2
test course	1	2
software testing method	1	2
set test cases	1	2
software testing execution	1	2
testing vehicle turn interaction term mechanics	1	2
testing process	2	2
engineering testing culture	1	2
exploratory testing	2	3
test system	1	3
simple obstacle avoidance test	1	3
test data	2	3
regarding software testing	2	3
testing methods	2	3
test cases	2	4
test code	1	4
physical testing	3	6