Software Qual J (2019) 27:349–387
https://doi.org/10.1007/s11219-017-9396-0

Comparison of the FMEA and STPA safety analysis methods–a case study

Sardar Muhammad Sulaman[1] · **Armin Beer**[2] · **Michael Felderer**[3,4] · **Martin Höst**[1]

Published online: 4 December 2017

Abstract As our society becomes more and more dependent on IT systems, failures of these systems can harm more and more people and organizations. Diligently performing risk and hazard analysis helps to minimize the potential harm of IT system failures on the society and increases the probability of their undisturbed operation. Risk and hazard analysis is an important activity for the development and operation of critical software intensive systems, but the increased complexity and size puts additional requirements on the effectiveness of risk and hazard analysis methods. This paper presents a qualitative comparison of two hazard analysis methods, failure mode and effect analysis (FMEA) and system theoretic process analysis (STPA), using case study research methodology. Both methods have been applied on the same forward collision avoidance system to compare the effectiveness of the methods and to investigate what are the main differences between them. Furthermore, this study also evaluates the analysis process of both methods by using a qualitative criteria derived from the technology acceptance model (TAM). The results of the FMEA analysis were compared to the results of the STPA analysis, which were presented in a previous study. Both analyses were conducted on the same forward collision avoidance system. The comparison shows that FMEA and STPA deliver similar analysis results.

Keywords Hazard analysis · Safety analysis · Critical systems · Failure mode and effect analysis · System theoretic process analysis

✉ Sardar Muhammad Sulaman
Sardar@cs.lth.se

[1] Department of Computer Science, Lund University, Lund, Sweden

[2] Beer Test Consulting, Baden, Austria

[3] Department of Computer Science, University of Innsbruck, Innsbruck, Austria

[4] Department of Software Engineering, Blekinge Institute of Technology, Karlskrona, Sweden

1 Introduction

The increasing dependence of our society on IT systems brings not only new development opportunities but also new severe risks and threats. As our daily life is almost completely dependent on IT systems, both for individuals and for organizations (private and public), failures of these IT systems can have serious negative consequences and effects on the society. In-depth and a completely performed risk and hazard analyses help to minimize the potential harm of IT system failures on the society (Leveson 2012; Sulaman et al. 2013). However, risk/hazard analyses of modern socio-technical systems are far from trivial, mainly due to the dynamic behavior that pervades almost every modern software-intensive system and a high number of interacting components. As a result, many traditional low-level risk or hazard analysis methods fail to encompass the dynamic behavior of the systems, as they focus solely on the system component failures (Leveson 2012). These traditional methods mainly focus on identification of critical components of a system and then either try to prevent the failures of these components or add redundant components. In case of dynamically changing systems, a new risk can emerge from wrong or non-synchronized commands that may lead to severe accidents. Therefore, new methods for performing risk and hazard analysis, optimized for dynamic systems, are required.

There are still a number of uncertainties when it comes to what risk and hazard analysis method to apply in a given situation. The main objective of this study is to empirically compare two existing risk analysis methods, failure mode and effect analysis (FMEA) and system theoretic process analysis (STPA). The study compares the results of and investigates the effectiveness of the well-established bottom-up FMEA and the rather new top-down STPA hazard analysis methods by performing a comparison of how a hazard analysis is conducted for the same system.

FMEA is a bottom-up analysis method that is used to identify potential failure modes with the causes for all the parts in system to find negative effects (I.E.C. 60812:2006 2006; MIL-STD-1629A 1980). The analysis starts with the lowest level components and proceeds up to the failure effect of the overall system. The main purpose of FMEA is to identify potential problems in the early design process of a system or product that can affect its safety and performance, and to introduce countermeasures to mitigate or minimize the effects of the identified potential problems (failure modes). On the other hand, STPA is a top-down method, just like the fault tree analysis (FTA) method. However, STPA uses a model of the system that consists of a functional control diagram instead of a physical component diagram used by traditional hazard analysis methods. It focuses on analyzing the dynamic behavior of the systems and is intended to provide advantages over traditional hazard analysis methods (Leveson et al. 2012). STPA is based on system theory unlike FMEA, which is based on reliability theory as explained in Sections 2.1 and 2.2. Moreover, STPA considers safety as a system's control (constraint) problem rather than a component failure problem.

The results of the FMEA analysis yielded from this study are compared with the results of a previous study (Sulaman et al. 2014) that presents an STPA hazard analysis of a system. It should be noted that this study does not aim at comparing both methods quantitatively, but instead to understand the differences through a qualitative analysis. That is, we investigate both methods qualitatively by analyzing hazard analysis results gathered by applying both methods, FMEA and STPA, on a collision avoidance system. Furthermore, this study also evaluates the analysis process of both the methods by using a set of qualitative criteria, derived from the technology acceptance model (TAM) (Davis 1989; Davis et al. 1989).

The remainder of this paper is structured as follows. Section 2 provides a background on FMEA, STPA, and other risk and hazard analysis methods, as well as an overview of

the forward collision avoidance system which is analyzed. Section 3 presents related work. Section 4 discusses the design of the case study and Section 4.5 presents the data collection procedure. Section 5 presents the results of the conducted analyses, Section 6 provides an analysis of the results, and Section 7 discusses the validity of the study. Section 8 discusses the results from the study, and Section 9 concludes the paper.

2 Background

This section presents a brief description of the FMEA and STPA hazard analysis methods that are compared in this study. It also provides an overview of other existing risk and hazard analysis methods. In addition, this section presents the description of the selected system, forward collision avoidance system, on which both methods are applied.

2.1 FMEA

FMEA is a bottom-up analysis method that is used to identify potential failure modes with the causes for all the parts in system to find negative effects (I.E.C. 60812:2006 2006; MIL-STD-1629A 1980). The analysis starts with the lowest level components and proceeds up to the failure effect of the overall system. A failure effect at a lower level becomes a failure mode of the component at the next higher level. FMEA also measures severity, occurrence, and detection probability that are used to calculate risk priority numbers for the identified failure modes. The main purpose of FMEA is to identify potential problems in the early design process of a system or product that can affect its safety and performance, and to introduce countermeasures to mitigate or minimize the effects of the identified potential problems (failure modes). Moreover, FMEA can complement FTA and identify many more failure modes and causes (McDermott et al. 2008). Failure modes and effects criticality analysis (FMECA) is an extension to FMEA that ranks the identified failure modes based on their severity, which is used for prioritization of countermeasures (Becker and Flick 1996; MIL-STD-1629A 1980).

Another extension is provided by Grunske et al. (2007) who introduced an extension to the conventional FMEA, namely "the probabilistic FMEA." It has the advantage of formally including rates at which component failures can occur. This method helps safety engineers to formally identify if a failure mode occurs with a probability higher than its tolerable hazard rate.

Software FMEA (SFMEA) (Pries 1998) is an extension to system FMEA to analyze software-intensive system components, such as embedded real-time systems. FMEA was originally aimed at the reliability of hardware. However, its benefits for performing a software FMEA were also shown by Stadler and Seidl (2013). Software FMEA considers specific aspects of software in an FMEA, for instance the fact that software components often do not fail in the traditional way but instead result in incorrect behavior. Software FMEA is a preventive measure for risk management and should therefore be carried out during the development of a system. Schmittner et al. (2014) state that SFMEA is best suited for a qualitative high-level analysis of a system in the early design phase. A general limitation of the FMEA analysis is the restriction to analyze only single cause of an effect. By assessing the severity of failure effects, the probability of their occurrence, and the detection of the probability of failure causes, a distinction between components of high or low risk is feasible and appropriate actions can be planned.

FMEA is applied to components in the design phase of the software system life cycle. The level of abstraction can take the levels of the V-model into account (Menkhaus and Andrich 2005). In this study, both software FMEA and system FMEA were applied in the following five steps:

1. Partition the system to be examined into subsystems and components, taking the architecture of hardware and software into account.
2. Assign the application function to each component. In this step, functional and non-functional requirements have to be interpreted.
3. Determine and analyze the potential failure mode, cause of failure, and failure effect that can lead to a hazardous state. For instance, the failure mode "false break activation" could have the cause of a defect in the SW of pressure determination and the effect of a potential crash situation between two cars. Another example regarding security is, for instance, a failure or threat mode classifying the way in which vulnerabilities are exploited (Schmittner et al. 2014). A threat mode could be "attacker is pretending to be a measurement device" violating the integrity of the system. The cause could be an encryption problem or security breach and results in "system is unreliable and potentially unsafe."

   Each failure mode represents potential product failures that can occur. Failure mode, cause, and effect are entered in the spreadsheet fields related to the appropriate component and function. The causal factors are associated with software defects, interface errors (architectural, protocol), HW/SW interaction (signaling), reliability, security, and real-time constraints. The potential failure effects could be the following: risk of collision, the operator is not alerted, a potential crash situation, or the authorization of external hackers to manipulate the collision avoidance system.
4. Evaluate risk and calculate the risk priority number (RPN). To calculate the RPN as described by McDonald et al. (2008), the severity of the failure effect, the probability of their occurrence, and the detectability of the failure causes have to be assessed first.

   The abbreviations used below for severity, probability, and detectability, i.e., B, A, and E, are adapted from the study (Mäckel 2001).

   - Severity (B): The severity value is assessed taking the potential failure effect into account. A five-point Likert scale is used, ranking the impact from 1 (no impact) to 5 (catastrophic, i.e., potential crash situation)
   - Probability of occurrence (A): To assess the probability of occurrence, the complexity, the potential failure mode, and cause of a failure have to be taken into account. A five-point Likert scale is used to rank the probability, starting from 1 (very low, 0.01%) to 5 (very high, 50%).
   - Detectability (E): The detectability depends on the complexity of the HW/SW component and potential cause of a failure. A five-point Likert scale is used to rank the detectability, starting from 1 (very low probability (0 to 19%) that current controls will detect the cause) to 5 (very high probability (80 to 100%) that current controls will detect the cause).
   - Calculation of risk priority number: RPN is calculated by multiplying the values of severity, probability of occurrence, and detectability. $RPN = B \times A \times E$, where B, A, and E denote severity, probability, and detectability according to above. RPN ranges from 1 to 125.
5. Specify defect avoidance or risk mitigation measures. This step is not taken into account in the current case study.

SFMEA (McDonald et al. 2008) allows the categorization of components taking the degree of their failure risk into account. It fosters the risk-oriented development of software-intensive systems. The complexity of a software system plays an important role in the development and the maintenance of products. SFMEA relates the complexity of a component to the probability of a failure. The practical experience in large-scale system development of the second author shows that if requirements are adapted iteratively, the complexity of the affected software components increases. In case also the system architecture has to be altered, the complexity will even increase significantly. SFMEA enables the partitioning of components into sets of different complexity. It considers complexity as an important influence factor in a hazard analysis. For example, a developer who focuses on the implementation of specific functions may overlook relations in the architecture of the system and therefore insert software defects. The benefit of FMEA is that complexity is taken into account to assess the risk of a failure and to issue preventive and analytical quality assurance measures like software testing (Felderer and Schieferdecker 2014).

2.2 STPA

The STPA method for hazard analysis focuses on analyzing the dynamic behavior of the systems and is intended to provide advantages over traditional hazard analysis methods (Leveson et al. 2012). STPA is a top-down method, just like the FTA method presented in Section 2.3. However, STPA uses a model of the system that consists of a functional control diagram instead of a physical component diagram used by traditional hazard analysis methods. STPA is based on system theory unlike FMEA, which is based on reliability theory. Moreover, STPA considers safety as a system's control (constraint) problem rather than a component failure problem. Among the most prominent benefits of STPA, Ishimatsu et al. (2010) listed the efficiency of the later phase of STPA when the broader scenarios are analyzed. According to Ishimatsu et al. (2010), STPA takes into consideration the interactions of system components and considers the evaluated system and its components as a collection of interacting control loops (control action and safety constraints on the component behaviors). STPA requires a control structure diagram for hazard analysis consisting of components of a system and their paths of control and feedback, i.e., acknowledgment. STPA is applied in the following two steps:

1. Identify the potential for inadequate control of the system that could lead to a hazardous state. A hazardous state is a state that violates the system's safety requirements or constraints and therefore can cause some loss regarding life, mission, or financial.
2. Determine how each potentially hazardous control action, identified in step 1, could occur (finding causal factors). An inadequate control action can lead a system to a hazardous state, and that could be one of the following:
   - A control action required is not provided.
   - An unsafe (incorrect) control action is provided.
   - A control action is provided too early or too late (wrong time or sequence).
   - A control action is stopped too early or applied too long.

The aforementioned term "provided" means the correct delivery of a control action or command from one component to another component of the system. A control action or command can encounter communication errors, e.g., delayed, failure, and corrupted. For the application of STPA, a functional control structure diagram of the system is required

and all control loops in system are identified from it. After this, in each control loop, all components that contribute to unsafe behavior of the studied system are identified.

Sulaman et al. (2014) applied STPA on a socio-technical system that has three controllers. They are critical components of system because they contain a process model (Leveson et al. 2012). The controller receives input from almost all components of the system, e.g., sensors and actuators, and then it performs internal calculations to issue a command.

2.3 Other methods

A few more risk and hazard analysis methods exist in addition to FMEA and STPA. For example, there exist a number of low-level risk analysis methods that analyze systems and subsystems at lower level considering only systems and their components. Some of the most well-known methods are fault tree analysis (FTA) (Ericson 1999) and hazard and operability study (HAZOP) (Redmill et al. 1999).

FTA is a top-down hazard analysis approach. It is a deductive approach and carried out by repeatedly asking: how can this (a specific undesirable event) happen, and what are the causes of this event? It involves a logical diagram that shows the relation between the system components and their failures. Ericson (1999) presented a review of the research performed on FTA with its advantages and shortcomings. Because FMEA is restricted to analyze only a single cause of an effect, FTA augments the feasibility of FMEA. An analysis using FTA in combination with FMEA may support an assessment considering, for instance, all security risks (Schmittner et al. 2014).

HAZOP is a qualitative technique commonly used in the planning phase of system development. It identifies hazards by analyzing how a deviation can arise from a design specification of a system. It is used to identify the critical aspects of a system design for further analysis. It can also be used to analyze an operational system. A multi-disciplinary team of 5–6 analysts lead by a leader usually carries out the HAZOP analysis. The HAZOP team identifies different scenarios that may result in a hazard or an operational problem, and then their causes and consequences are identified and analyzed (McDermid et al. 1995).

2.4 Forward collision avoidance system

The forward collision avoidance (FCA) system was selected in this study to compare and evaluate hazard analysis methods. Here, it should be noted that the main focus of this study is on the comparison and evaluation of the analysis methods (FMEA and STPA) rather than the FCA system itself. The FCA system was selected because it was decided to use an operational and real system for the analysis. Here, we have tried to find a system to be analyzed that is representative of systems suitable for both methods.

The FCA system alerts a driver of a vehicle about crash situations and applies automatic brakes after a certain time period if the driver does not respond to a warning alert that provides passive and active safety. The system performs two main functions: (1) object/obstacle detection (by using forward-looking sensors that detect hindrance in front of the vehicle) and (2) generation of warning or applying auto breaks (passive/active response). The forward-looking sensors could have some or all of these components: radar, infrared, motion sensors, and cameras (Bond and et al. 2003; Coelingh et al. 2010).

Figure 1 shows the forward collision avoidance system adapted from Bond and et al. (2003). Here, it has been divided into three parts such as parts A (the collision controller), B (the brake controller), and C (the engine torque controller).

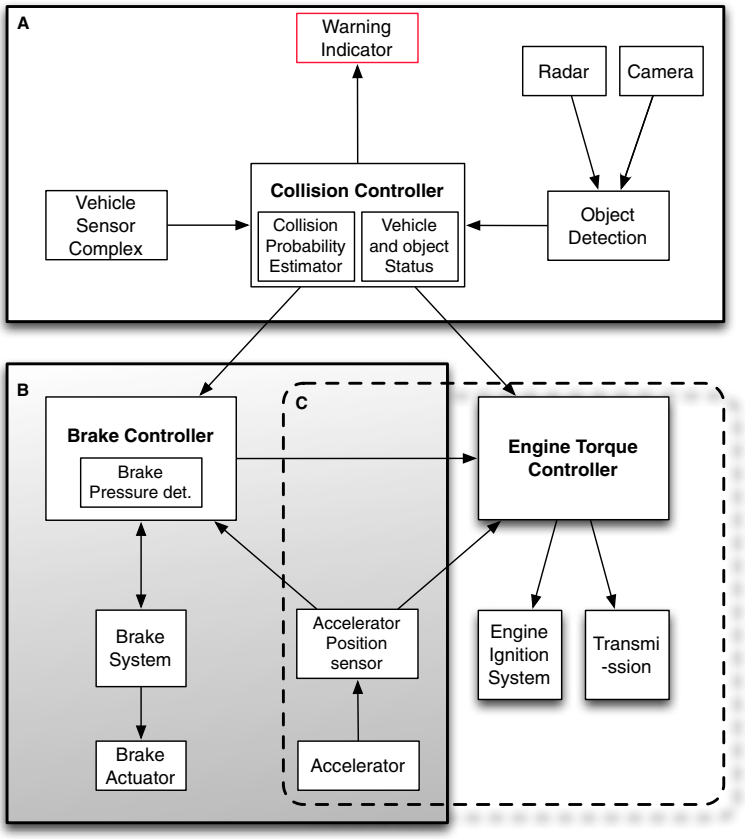

Fig. 1 Forward collision avoidance system with autonomous braking (Bond and et al. 2003)

The *collision controller* (part A of the system) is connected with the following system components: The *collision controller* is connected with the *radar* and the *camera* through the *object detection system*. An object detection system could have more sensors or devices to detect an object in front of the vehicle. In this study, we suppose that it uses more than one motion sensors to complement the radar and the camera. The object detection system could be very simple or very complex but in this study, we consider the simple version. In the next sections, we will only refer to the object detection system instead of referring individually to the radar, camera, and sensors.

The *vehicle sensor complex* is also connected with the collision controller that generates a signal and then sends it to the collision controller. The vehicle sensor complex consists of several vehicle system sensors, such as a brake position sensor, throttle position sensor, steering sensor, suspension sensor, speed sensor, and seat belt sensor. The information from these sensors can either be used individually or together to complement the collision avoidance system.

The *warning indicator* connected with the collision controller generates a collision warning signal in response to the collision assessment of the collision controller. The collision controller gets input from the object detection system and the vehicle sensor complex when it performs the collision assessment.

The *collision controller* (shown in part A) works as follows: The *vehicle* and *object* status provider in the collision controller calculates and provides the current status of the object in front of the vehicle and the current status of the vehicle to the collision probability estimator. The *collision probability estimator* in the collision controller calculates the vehicle collision probability based on the received information. If there is a risk of collision, then the estimator sends a signal to the indicator, which is for the vehicle's operator. This is known as collision detection, which is a passive safety system that just warns the vehicle operator. If the vehicle operator does not respond to the collision warning, then the system activates the collision avoidance system also known as the active safety (autonomous brake). The *collision controller* uses an algorithm to estimate the risk of collision and generates a collision-assessment signal. It is a critical component of the collision avoidance system, because both active safety and passive safety depend on the output of this component. It also calculates some other parameters, such as the time to collision that is going to happen, point of collision, and object identification. If the vehicle's operator responds to the collision warning on time, then the forward collision avoidance system resets all its components and calculated parameters. However, if the operator does not respond to the received warning, then the collision controller sends a collision-assessment signal with the object and vehicle status signals to the *brake* and *engine torque* controllers to apply autonomous brake.

The *brake controller* (part B of the system) works as follows: It receives the *vehicle status* signal, *detected-object status* signal, and *collision-assessment* signal from the *collision controller*. The brake controller has one *brake pressure measurement or determination* component that determines the required brake pressure for the current situation based on the received information from the *collision controller* and *accelerator position sensor*. After determining the required brake pressure, the brake controller sends an autonomous brake signal to the *brake system* and to the *engine torque controller*. The *brake system* has one *brake pedal* and one *brake actuator* that apply the autonomous brakes. One important action of the brake controller and brake system is that they allow the vehicle's operator intervention during the application of autonomous braking. Operator can increase the brake pressure by intervening the autonomous braking that also deactivates the collision avoidance system in that particular collision situation. The *engine torque controller* (part C of the system) works as follows: It reduces the torque to almost zero after receiving signals from the *collision controller* and *brake controller* during the application of autonomous braking by using different methods like by limiting air or fuel supply to engine, downshifting the transmission, and switching the engine off. The *accelerator position sensor* is electrically coupled to the brake controller and the engine torque controller that indicates and provides the position of accelerator.

2.5 Hazard

The term "hazard" used in this study generally defined as "anything that has the potential to do harm" or "anything that can lead to an accident." According to Leveson (2012), if every state of a system is considered, then system can always pose a potential danger or itself in danger. Therefore, this definition should preclude states that the system must normally be in to accomplish the mission. However, this study is not trying to define a new definition for "hazard" instead it follows the general definition adapted by Leveson (2012).

> "A system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to an accident (loss)."

3 Related work

The research objective of the current study is to compare the STPA and FMEA hazard analysis methods. There exist some studies that have compared different risk and hazard analysis methods. For example, Stålhane and Sindre (2007) performed a comparison of two safety analysis methods, misuse case (MUC) method and FMEA. The MUC method was originally proposed for eliciting security requirements (Sindre and Opdahl 2005), but it has also been used for safety analysis. The MUC method was developed by the software community as an alternative to FMEA and HAZOP. Both methods were compared in an experiment to investigate which method is better than the other for identifying failure modes and if one of the methods was easier to learn and to use. The authors concluded that when the system's requirements are described as use cases, MUC is better than FMEA for analyzing failure modes related to user interactions. Furthermore, FMEA is better than MUC for analyzing failure modes related to the inner working of the system. The authors also concluded that MUC will create less confusion and in general be easier to use than FMEA.

Yu et al. (2011) compared and discussed three well-known risk analysis methods by applying them on a box fan: FMEA, advanced failure mode and effect analysis (AFMEA) (Eubanks et al. 1997), and FTA. The authors presented the advantages and disadvantages of these methods and concluded their study with an attempt of combining both deductive (top-down) and inductive (bottom-up) risk/safety analysis methods.

Ishimatsu et al. (2014) compared the STPA hazards analysis results with the FTA analysis results that were used to certify the H-II Transfer Vehicle (HTV). The HTV is an unmanned cargo transfer spacecraft that is launched from the Tanegashima Space Center aboard the H-IIB rocket and delivers supplies to the international space station (ISS). In the development of the HTV, the potential HTV hazards were analyzed using FTA and during the analysis, the NASA safety requirements were also considered. After comparison of the results, the authors concluded that STPA identified all the traditional causes of losses identified by FTA and FMEA, but it also identifies additional causes. The additional factors include those that cannot be identified using fault tree analysis, including software and system design as well as system integration.

Fleming et al. (2012, 2013) analyzed the NextGen In-Trail Procedures (ITP) application by using the STPA analysis method and compared its results with the official NextGen ITP application analysis (RTCA/DO-312 2008). NextGen is the next generation of air traffic management systems that contains In-Trail Procedures application. ITP is an application of Automatic Dependent Surveillance-Broadcast (ADS-B) that allows aircraft to change flight levels in areas where current radar separation standards would prevent desirable altitude changes (Haissig and Brandao 2012). To summarize, ITP helps to increase operational efficiency and throughput in oceanic airspace (Fleming et al. 2013). The authors concluded that STPA found more potential causes of the hazards considered (violation of separation requirements) than the traditional hazard analysis performed on ITP (RTCA/DO-312 2008). In the comparison, the authors identified 19 safety requirements that were not in either of the two official NextGen analysis documents.

Fleming et al. (2012, 2013) also compared STPA with bottom-up and other top-down analysis techniques. According to the authors, bottom-up analysis techniques, FMEA, start by identifying all possible failures. This list can be very long if there are a lot of components and all the permutations and combinations of component failures are considered. However, STPA only identifies the failures and other causes that can lead to a system hazard and does not start by identifying all possible failures. Moreover, in the top-down STPA analysis approach, the analyst can stop refining causes at the point where an effective

mitigation can be identified and does not go down any further in detail. The analyst only has to continue refining causes if an acceptable mitigation cannot be designed. That is the major difference between STPA and FMEA (and any other bottom-up technique), which explains the differences in time and effort required (Fleming et al. 2012; 2013).

Furthermore, Nakao et al. (2011) evaluated STPA in a case study where it was applied on an operational crew-return vehicle design. The authors conclude that with STPA, it is possible to recognize safety requirements and constraints of the system before the detailed design.

Raspotnig and Opdahl (2013) compare risk identification techniques for safety and security requirements. From the safety field, the functional hazard assessment (FHA), the preliminary hazard analysis (PHA), HAZOP, and FMEA as well as FTA are considered. Each technique is assessed based on several quality criteria addressing the context, the application area, and the application method as well as advantages and disadvantages of utilizing the technique. The assessment is based on evidence reported in the literature. The authors conclude that risk identification techniques for safety are more mature than for security and that they have found a balance between creativity and formalism, which is needed for identification process.

As it can be noticed from the literature mentioned in this section, STPA is a quite new analysis technique as compared to other techniques (FMEA, FTA, etc.). Fleming et al. (2012, 2013) mention that traditional analysis methods (FMEA, FTA, etc.) are more than 50 years old and, while analyzing safety critical software-intensive systems, they cannot identify software faults or the errors pertaining to dynamic behavior of the system. SFMEA (Pries 1998) and especially also STPA (Leveson et al. 2012) have been developed to overcome the existing problems in traditional analysis methods. According to Leveson et al. (2012) and Fleming et al. (2012, 2013), STPA can find more component interaction, software, and human hazards than traditional methods. Therefore, according to the authors, STPA is more effective because it is developed by considering system thinking that considers whole system as a single unit and finds more hazards. Moreover, previously, STPA is compared and evaluated with bottom-up methods (e.g., FMEA) by the same authors who presented it or they were involved in its development. Several authors (Leveson 2012; Pereira et al. 2006; Thomas and Leveson 2011; Ishimatsu et al. 2010; Nakao et al. 2011; Fleming et al. 2013) reported positive outcomes from applying STPA on various systems. However, the traditional methods are still in use in practice even though they are more than 50 years old for the analysis of safety critical systems in early design, development, and operational phases. This means that there is a need for further investigation of effectiveness of the STPA method compared to other traditional safety analysis methods that are used in industry. If further investigations find STPA as an effective method, then these results can help industry to shift to this new analysis method.

To summarize, it is interesting to investigate what are the main differences in STPA and other traditional methods (in this case FMEA) and also the types of hazards identified by them.

4 Case study design

4.1 Research objective

The main objective of this study is to compare and investigate effectiveness of FMEA and STPA hazard analysis methods in the software-intensive safety-critical system domain. In

this study, hazard analysis results from both FMEA and STPA are compared to find the main differences in methods by investigating, e.g., types of hazards identified by them. Based on the comparison results, this study also investigates which method is more effective. Moreover, this study also evaluates the analysis process of both methods by using a qualitative criteria derived from the technology acceptance model (TAM).

4.2 Research questions

The aforementioned research objective has been broken down in the following main research questions.

RQ1: What are the main differences between the selected hazard analysis methods regarding types of the identified hazards?
RQ2: What are the main differences in the analysis process of both methods?
RQ3: Which method is more effective, FMEA or STPA?

In our context, effectiveness is high if a large number of relevant hazards but only a small number of non-relevant hazards are identified.

RQ1 is answered by analyzing and investigating the results from both the FMEA and STPA analyses to find the main differences between the two methods. Five error types were defined based on the related studies (Leveson et al. 2012; Fleming et al. 2012, 2013) and then all the identified hazards are classified according to the defined error types. Furthermore, the classification of error types (identified hazards) is investigated to answer which method finds what types of hazards.

RQ2 is answered by developing the qualitative criteria to evaluate the analysis process of both methods. The qualitative criteria were derived from the TAM to evaluate the analysis process considering ease of use and usefulness. Then, the developed qualitative criteria were applied on both methods to analyze and evaluate them.

RQ3 is answered by analyzing and investigating the results from both the FMEA and STPA analyses. It should be noted that, in this research, initiative hazard analysis of collision avoidance system is carried out using only the FMEA method. After this, the FMEA results are compared to the STPA results found in a previous study (Sulaman et al. 2014).

4.3 Research methodology

In this study, the FMEA hazard analysis method is applied on the forward collision avoidance system in order to compare the results with the results of the previous study (Sulaman et al. 2014). In the previous study, the STPA hazard analysis method was applied on the same system in order to understand more about STPA and assess its effectiveness and efficiency. Here, we have tried to use a system for analysis that is representative of systems suitable for both methods. The steps carried out for the presented research in this study are shown in Fig. 2.

Step 1 denotes the steps carried out in the previous study (Sulaman et al. 2014), where the first author of the current study analyzed the forward collision avoidance system and identified inadequate control commands or events. After this, the identified inadequate control commands or events were analyzed for their causal factors.

In step 2, the second author of this study applied the FMEA method on the same collision avoidance system to analyze operational hazards (hazards that endanger the safety of the system, when it is operated) in it. The first author already knew the existing hazards in the selected system because he had applied STPA on the selected system in the previous study

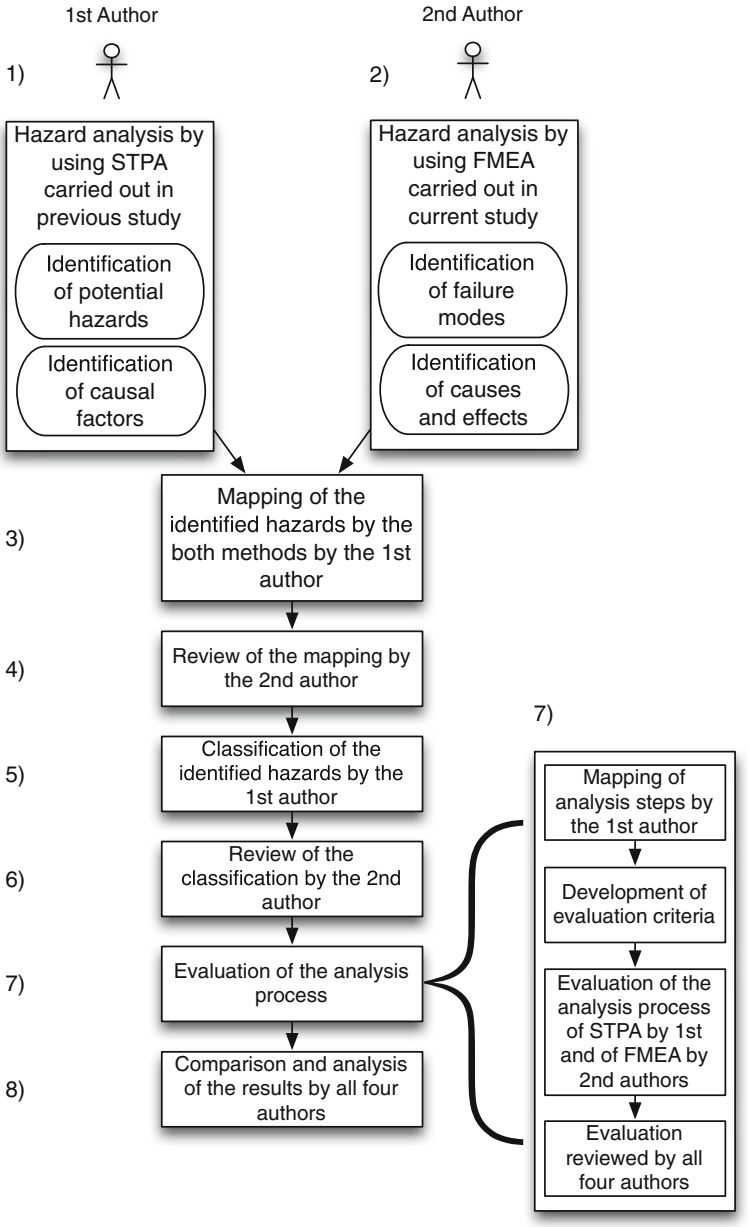

Fig. 2 Steps taken for the carried out research

(Sulaman et al. 2014). Therefore, to improve the research validity, it was decided that the first author would not apply the FMEA method; instead, the second author would carry out FMEA analysis as he has experience of analyzing safety critical systems. For the FMEA analysis, all the documents about the selected system description that were used during the analysis in the previous study (Sulaman et al. 2014) were provided to the second author of

this study to apply FMEA. During the FMEA analysis carried out in this study, a number of measures were taken in order to increase the research validity and to decrease researcher bias. The same system information and description were available to the second author to analyze the system, as were used in the previous study, and the second author of this study did not review the previous study results.

Regarding the time spent on analysis in steps 1 and 2, the first and second authors were allowed to spend as much time as required for the analysis, since the objective was to perform a detailed analysis that can be used for the qualitative comparison of both methods. The required time was not measured with great precision, since this was not a main research question in the research. However, the time required for both methods was subjectively estimated after the process, and they were about the same. At least, there were no major differences

After this, in step 3, the first author of this study performed an initial comparison (mapping) of the identified hazards yielded from the FMEA analysis and STPA analysis. The first author created a list of the common hazards that were identified by both analysis methods and another list was created for the distinct hazards identified by only FMEA or STPA.

Then, in step 4, the second author reviewed the initial comparison performed by the first author. The second author identified one more hazard (no. 18 in Table 4) as a common hazard.

In step 5, the first author classified all the identified hazards into the following five error categories: component interaction error, software error, human errors, component error, and system error. These categories were selected because the STPA (Leveson et al. 2012; Fleming et al. 2012, 2013) method claims to identify these types of hazards. According to these sources, STPA can find more component interaction, software, and human hazards, which could be investigated by classifying the hazards in this way.

After this, in step 6, the second author reviewed the classification performed by the first author, and additionally, the third and fourth authors reviewed the results of steps 1 to 6.

Then, in step 7, all four authors had a discussion regarding the development of qualitative criteria to evaluate the analysis process of both methods. After this discussion, the criteria were developed and then the first author mapped analysis steps of both methods. Then, the first author evaluated STPA and the second author evaluated FMEA according to the developed evaluation criteria. After this, the third and fourth authors reviewed the mapping and evaluations performed by the first and second authors.

Finally in step 8, a final comparison and analysis was performed by all authors to investigate the differences between both methods.

4.4 Case and unit of analysis

As the objective of this study is to evaluate and compare hazard analysis methods, the case for this study is a composite case that consists of a risk analysis method and a system on which the method is applied to analyze hazards. The selected case for this study is the FMEA and STPA risk analysis methods along with collision avoidance system. A similar case study was carried out in an earlier study (Sulaman et al. 2014) that also contains a composite case consisting of a hazard analysis method, STPA, and a system, collision avoidance system, on which method was applied to analyze hazards. However, in this study, the case consists of both methods and the analyzed system because the objective is to evaluate and compare both methods based on the results yielded from the analyses.

4.5 Data collection procedures

In this study, the system description along with the system control structure diagram that shows how it works is used as study objects for the hazard analysis and evaluation of the methods. The system description is gathered from the existing patents for collision avoidance systems and also from the published literature for collision avoidance system (Bond and et al. 2003; Coelingh et al. 2010). Moreover, data collected through the hazard analysis from both methods is also used for the analysis and evaluation of the methods. Besides this, the expert opinions and knowledge are also used to evaluate and investigate the performed hazard analyses for analysis and results.

5 Results

5.1 Safety analysis using FMEA

The risk analysis using FMEA is performed to view the occurrence of failures in the collision avoidance system in a preventive manner. Table 1 shows the failure mode and effect analysis for the collision avoidance system. The FMEA was performed by the second author of this study according the procedure described in Section 2.1.

The first three columns show the identified subsystems, components, and functions, for instance FMEA No. 9 the brake actuator, which switches from the auto brake to the manual brake, in case of a failure (corresponding to steps 1 and 2). The fourth column (Potential failure mode(s)) shows the failure mode, i.e., "Activation of manual brake fails," corresponding to step 3. Each failure mode is taken to the potential causes (column 5) and effect of a failure (column 6). For instance, focusing on FMEA No. 9, the cause may be a software defect of handling events or queues; as a consequence, improper brake activation and a potential "crash" with an adjacent vehicle may occur. Each failure mode would be a hazard for a safe usage of the product.

In step 4, the worst-case impact of the effect of a failure, i.e., severity, the probability of occurrence, and detectability are assessed. For example, for FMEA No. 9, the severity is 5, because of a potential "crash" situation; the probability of occurrence is 4 (high), because the complexity of the HW/SW component is very high; and the likelihood of a failure is 10%. Its detectability is 4 (low), because the probability, that current reviews and testing of artifacts will detect the defect, is 20–39% only.

The RPN is, as described above, calculated by multiplying the values of severity, probability of occurrence, and detectability, $RPN = B \times A \times E$, which means that it ranges from 1 to 125. In the case of FMEA No. 9, the $RPN = 80$, which is the highest value shown in Table 1. To mitigate the risk of a software failure in the first place (step 5), extensive functional tests and code reviews are performed in the development phase. However, the RPN value is in some cases misleading. For instance, in FMEA No. 21, the effect of the failure mode "attacker is pretending to be a measurement device" is defined as "system is unreliable and potentially unsafe," the severity is 5, but the probability of occurrence is very low (0.01%), the detectability is 3 (medium probability), and the $RPN = 15$ only. Lower detectability, however, results in more risk and is therefore ranked higher. An intrusive attack of the collision avoidance system has to be blocked, for instance by encryption of signals, to mitigate the risk of manipulation of the braking or engine torque controller system. Security tests based on attack patterns are performed during development to mitigate the risk of a security breach in the system.

Table 1 Failure mode and effects analysis (quality risk analysis)

No.	System	Component	Application function	Potential failure mode(s)	Potential cause of failure	Potential failure effect	Risk assessment				Risk mitigation measures
							B	A	E	RPN	
1	Part A: collision controller system	Collision probability estimator	Calculation of probability of collision	Erroneous probability estimation	SW-defect: calculating probability or in taking environment constraints into account.	Risk of collision with vehicle in front	4	4	5	80	Funct. test-strength = 3, code review
2		Vehicle and object status	Calculation of speed and position of object	Erroneous calculation of speed of object	Erroneous specification or implementation of state transitions.	Risk of collision with vehicle in front	4	4	4	64	Funct. test-strength = 3
3		Warning indicator	Man-machine interface	Display of warning fails	SW defect	Operator is not alerted about potential danger	3	3	2	18	Funct. test-strength = 2
4		Vehicle sensor complex	Steering of collision controller	Erroneous signaling of sensors	Architecture erroneous or missing services provided by sensors.	Operator is not alerted about potential danger	3	4	4	48	Funct. test-strength = 3, design review
5		Object detection	Object detection by radar or camera	Erroneous radar and camera	SW defect or failure: object detection devices.	Operator is not alerted about potential danger	3	3	2	18	Funct. test-strength = 3
6		Collision controller interface	Switch over to complement device	Object recognition device fails	SW defect or failure of communication stack.	Operator is not alerted about potential danger	3	3	4	36	Funct. test-strength = 3, code review

Table 1 (continued)

No.	System	Component	Application function	Potential failure mode(s)	Potential cause of failure	Potential failure effect	Risk assesment				Risk mitigation measures
							B	A	E	RPN	
7	Part B: brake controller system	Brake controller	Activation of brake system and engine torque controller	Steering of brakes fails	SW defect or failure of communication stack.	Auto brake is not activated properly, risk of collision	4	3	2	24	Funct. test-strength = 3
8		Brake pressure determination	Calculation of brake pressure	Steering of brakes fails	SW defect in pressure determination.	Auto brake is not activated properly, risk of collision	4	3	5	60	Funct. test-strength = 3, code review
9		Brake system	Switch over from auto brake to manual brake	Activation of manual brake fails	SW defect: handling events or queues.	Improper brake activation, potential crash situation	5	4	4	80	Funct. test-strength = 3, code review
10		Brake system	Switch over from manual brake to auto brake system	Activation of auto brake fails	SW defect: handling events or queues.	Improper brake activation, potential crash situation	5	4	4	80	Funct. test-strength = 3, code review
11		Engine torque controller interface	Deactivation of torque controller	Erroneous torque controller is still active	SW defect or missing services provided: engine torque controller	Operator is not alerted about potential crash situation	4	3	3	36	Funct. test-strength = 3, code review
12		Collision controller interface	Deactivation of collision controller	Automatic collision avoidance system is still running, when it should stop.	SW defect: activation or deactivation of collision avoidance system	Operator is not alerted about potential crash situation	4	2	3	24	Funct. test-strength = 3, code review

Table 1 (continued)

No.	System	Component	Application function	Potential failure mode(s)	Potential cause of failure	Potential failure effect	Risk assesment				Risk mitigation measures
							B	A	E	RPN	
13	Part C: engine torque controller system	Collision controller interface	Reception of data from collision controller	Vehicle and object status determination fails.	Architecture erroneous or missing services provided by components, sensors.	Operator is not alerted about potential crash situation	4	3	2	24	Funct. test-strength = 2, design review
14		Brake controller interface	Reception of data from braking system	Brake pressure determination fails	SW defect or missing services provided: interface, sensors.	Operator is not alerted about potential crash situation	4	3	2	24	Funct. test-strength = 2
15		Accelerator position sensor and accelerator	Calculation of acceleration and position	Erroneous interpretation of sensor signals	SW defect: processing of sensor data.	Operator is not alerted about potential danger	3	3	4	36	Funct. test-strength = 3, code review
16		Engine ignition system	Steering of engine ignition	Engine ignition is not activated	SW defect: steering of engine ignition.	Improper stop of vehicle and potential crash situation.	5	2	4	40	Funct. test-strength = 3, code review
17		Transmission	Steering of transmission	Transmission downshifting fails.	SW defect or failure of communication stack.	Potential crash situation	5	2	4	40	Funct. test-strength = 3
18		Engine torque controller	Torque reduction	Torque is not reduced	SW defect: state recognition	Operator is not alerted about potential danger	3	2	4	24	Funct. test-strength = 3
19	Parts A, B, C	Operating system	Communications	Bad performance	SW defect: handling events or queues.	Potential crash situation	5	2	4	40	Performance test-strength = 3
20		SW/HW components	Processing	Frequent restarts of the system	Sporadic SW defects	System is no longer reliable	4	3	5	60	Stress test-strength = 3
21		Trust boundaries	Protecting system assets	Attacker is pretending to be a measurement device	Encryption problem or security breach.	System is unreliable and potentially unsafe.	5	1	3	15	Security test with attack pattern

To summarize, to reduce the occurrence of probability, error preventive measures during development, such as coding guidelines, should be used. To mitigate the risk of a software failure during operating, code reviews as well as functional, performance, and security tests are performed during development. The intensity of testing takes the complexity of the components, the severity, the probability of the occurrence, and the effect of failure in respect of the safety and security of the system into account. About 71% (15) of all potential failures were identified as "catastrophic" or "critical," 29% (6) as moderate, and none as marginal failure. It can be noticed from Table 1 that potential causes of failures are software faults, erroneous HW/SW interfaces, or missing services. Thus, the majority of the identified hazards and their causes correspond in the first place to software faults, insufficient reliability, performance, and security. FMEA supports, similar to STPA, risk analysis. However, FMEA fosters also preventive measures during the development of a product or when the system is in operation. The quality of a complex embedded system is monitored by the interpretation of the FMEA, to issue defect detection measures before going into operation. To support also an efficient maintenance of the product, the FMEA worksheet should be updated regularly.

5.2 Safety analysis using STPA

For hazard analysis using STPA, the detailed control structure diagram of the system was acquired. Then, the first author of this study analyzed the forward collision avoidance system and identified inadequate control commands or events (for detail see Sulaman et al. (2014)). Table 2 shows the inadequate control commands or events that could lead to hazardous states. During step 1 of STPA, 14 inadequate control commands or events have been identified in the forward collision avoidance system. Then, these control commands or events were analyzed, one by one, to identify their associated hazards. As one can see from Table 2, not provided control commands lead the system under consideration to hazardous states, in most cases of catastrophic level. Similarly, all identified control commands or events provided too late lead to, in most cases, hazardous states of catastrophic level. On the other hand, none of the events provided too early lead to catastrophic hazardous states; three lead to moderate and one to negligible level hazards.

It can be noted that one hazard can have more than one inadequate control action, e.g., hazard 2a in Table 2 exists because of vehicle sensor complex signal is not provided, provided unsafe, and provided too late. For all these three inadequate control actions, there is a single hazard.

From the identified 14 inadequate control commands or events, 22 hazards were identified. Table 3 shows the causal factors for all identified hazards in step 1 with their severity levels. The first column of Table 3 shows the identified hazards, the next column shows the severity levels, and the third column shows the causal factors for all hazards. The hazards were classified in three severity levels: catastrophic, moderate, and negligible. Over 70% (16) of all the hazards were classified as catastrophic with potentially fatal consequences. Only three hazards were classified as moderate severity level that may lead to severe accidents and have risk of serious injury. The remaining three hazards have negligible severity level. The negligible hazards do not have any serious consequences if the pertaining component fails alone and the other components of the system work properly. Therefore, based on the results of Sulaman et al. (2014), it is possible to hypothesize that the STPA method efficiently supports risk analysts with limited domain experience (in our case maximum 5 years) in the identification of complete set of catastrophic hazards.

From Table 3, it can be noticed that the causal factors associated with component failures, communication errors, and software faults (dynamic behavior) were identified. Thus, the majority of the identified hazard and their causes correspond to the software faults of the studied system.

6 Analysis

This section presents the analysis of the results that encompasses the main comparison results of both methods, FMEA and STPA, and their evaluation results based on the developed criteria.

6.1 Common and distinct hazards identified by both methods

Table 4 shows the mapping of the hazards identified by both analysis methods. The identified hazards are represented in Table 4 by using their numbers used in Tables 1, 2, and 3. As it can be noticed from Table 1, the analysis by FMEA found 21 hazards. On the other hand, STPA found 22 hazards shown in Tables 2 and 3. In total, both analysis methods found 30 unique or distinct hazards. As shown in Table 4, there are some hazards identified by STPA which are on a more abstract level compared to corresponding hazards identified by FMEA. For example, hazards 2a and 12a (identified by STPA) are mapped to two hazards each identified by FMEA. As it can be noticed from Table 4, there are some identified hazards that are only identified by one analysis method, either FMEA or STPA. Table 4 and Fig. 3 also show 13 common hazards identified by both analysis methods.

6.2 Classification of the identified hazards

The identified hazards are classified into the following five error categories:

- Component interaction error
- Software error
- Human error
- Component error
- System error

These categories were selected because it is claimed that the STPA method identifies these types of hazards (Leveson et al. 2012; Fleming et al. 2012, 2013). Furthermore, this list of error types seems to be complete because it covers all system components. The traditional methods, FTA and FMEA, are more focused on components because they are based on reliability theory. Therefore, the most relevant error type for traditional methods is component failure. Software-intensive system is mainly consists of software, hardware, human, interaction between system components and human, and system itself. If we take these components of a software-intensive system, then the selected five error types cover all the components of system. Therefore, the authors believe that these five error types are sufficient to discuss and compare different analysis methods.

Figure 4 shows three bar plots showing classification for the common and distinct identified hazards by both methods. The first bar plot shows the hazards only identified by the STPA method. The second bar plot shows the hazards only identified by the FMEA method. Finally, the third bar plot shows the classification of the hazards identified by both methods (common hazards).

Table 2 Inadequate control actions/commands

No.	Command/event	Not provided	Provided unsafe	Provided			Stopped too soon
				Too early	Too late	Out of sequence	
1	Object detection signal	Catastrophic-system dysfunction [collision] (1a)	Catastrophic-system malfunctioning (1b)	N/A	Catastrophic-system dysfunction [collision] (1a)	N/A	N/A
2	Vehicle complex signal	Catastrophic-problem in calculation of vehicle status and collision probability (2a)	Catastrophic-problem in calculation of vehicle status and collision probability (2a)	N/A	Catastrophic-problem in calculation of vehicle status and collision probability (2a)	N/A	N/A
3	Collision warning signal	Negligible (if every thing is working properly, then the active safety will be saved from collision) (3a)	N/A	Negligible (if every thing is working properly, then the active safety will be saved from collision) (3a)	Negligible (if every thing is working properly, then the active safety will be saved from collision) (3a)	N/A	Negligible (warning will be stopped too soon that can cause accident. If everything works properly, then the active safety will be saved from collision) (3b)
4	System reset signal (response from driver by using brakes)	Negligible (if everything is working, then the active safety will be saved from collision) (4a)	Negligible (if everything is working, then the active safety will be saved from collision) (4a)	N/A	Negligible (if everything is working, then the active safety will be saved from collision) (4a)	N/A	N/A
5	Vehicle status signal	Catastrophic (wrong brake pressure determination) (5a)	Catastrophic (wrong brake pressure determination) (5a)	N/A	Catastrophic (wrong brake pressure determination and decrease in reaction time) (5a)	N/A	N/A
6	Object status signal	Catastrophic (wrong brake pressure determination) (6a)	Catastrophic (wrong brake pressure determination) (6a)	N/A	Catastrophic (wrong brake pressure determination and decrease in reaction time) (6a)	N/A	N/A

Table 2 (continued)

No.	Command/event	Not provided	Provided unsafe	Provided			Stopped
				Too early	Too late	Out of sequence	too soon
7	Collision assessment signal	Catastrophic-system will not work [collision] (7a)	Catastrophic-system will not work as intended [collision] (7b)	Moderate-false signal due to system malfunctioning [application of automatic brakes without need] (7c)	Catastrophic-system will not work [collision] (7a)	N/A	N/A
8	Reduce torque	Moderate-collision with divider, other things, and vehicle can slip (8a)	N/A	N/A	Moderate-collision with divider, other things, and vehicle can slip (8a)	N/A	N/A
9	Brake signal with required pressure	Catastrophic-system dysfunction [collision] (9a)	Catastrophic-system malfunctioning [collision] (9b)	Moderate-false signal due to system malfunctioning [application of automatic brakes without need] (9c)	Catastrophic-system dysfunction [collision] (9a)	N/A	N/A
10	Apply brakes signal	Catastrophic-system dysfunction [collision] (10a)	N/A	Moderate-false signal due to system malfunctioning [application of automatic brakes without need] (10b)	Catastrophic-system dysfunction [collision] (10a)	N/A	N/A
11	Accelerator signal	Catastrophic (wrong brake pressure determination) (11a)	Catastrophic (wrong brake pressure determination) (11b)	N/A	Catastrophic (wrong brake pressure determination) (11a)	N/A	N/A
12	Change transmission signal	Catastrophic-torque will not be reduced (12a)	N/A	N/A	Catastrophic-torque will not be reduced (12a)	N/A	N/A
13	Limit air and fuel supply signal	Catastrophic-torque will not be reduced (13a)	N/A	N/A	Catastrophic-torque will not be reduced (13a)	N/A	N/A
14	Switch off engine signal	Catastrophic-torque will not be reduced (14a)	N/A	N/A	Catastrophic-torque will not be reduced (14a)	N/A	N/A

Table 3 Causal factors of the identified hazards

No.	Step 1 no.	Hazards	Severity	Causal factors
1	1a	System dysfunction due to failure of object detection system	Catastrophic	Object detection component failure (camera, radar, or motion sensors)
				Communication error (no signal)
2	1b	Malfunctioning of the system due to incorrect input from object detection system	Catastrophic	Corrupted communication (wrong signal)
				Malfunctioning of camera, radar, and motion sensors
				Delayed communication (system will not work on time)
3	2a	Incorrect and missing calculation of vehicle status and collision probability due to failure or malfunctioning of vehicle complex sensors	Catastrophic	Failure of vehicle sensors
				Communication error (no signal)
				Delayed communication (system will not work on time)
				Malfunctioning of sensors (incorrect values sent by sensors)
4	3a	Missing collision warning signal-if rest of the system is working properly, then the active safety will be prevented from collision	Negligible	Inadequate collision assessment algorithm, failure of warning indicator
				Malfunctioning of warning indicator, incomplete controller process model
				Failure of collision estimator, malfunctioning of collision estimator
				Incorrect vehicle or object status, communication error (no signal)
				Delayed communication (system will not work on time)
5	3b	If warning stopped too soon, then it can cause accident-if everything else will work, then the active safety will handle the situation	Negligible	Failure of warning indicator
				Malfunctioning of warning indicator
				Communication error
6	4a	Missing system reset signal can cause collision with divider or other objects due to unwanted auto braking	Negligible	Brake pedal sensor failure
				Communication error (no signal)
				Delayed communication (system will not reset on time and will apply brakes)
7	5a	Incorrect brake pressure determination due to missing vehicle status signal	Catastrophic	Failure of vehicle sensor complex (2a)

Table 3 (continued)

No.	Step 1 no.	Hazards	Severity	Causal factors
				Malfunctioning of collision controller due to incomplete process model
				Communication error (no signal)
				Delayed communication (system will not work on time)
8	6a	Incorrect brake pressure determination due to missing object status signal	Catastrophic	Failure of object detection (1a)
				Malfunctioning of collision controller due to incomplete process model
				Communication error (no signal)
				Delayed communication (system will not work on time)
9	7a	System dysfunction due to missing collision assessment signal	Catastrophic	Component failures in object detection and vehicle complex signal (1a and 2a)
				Failure of collision probability estimator
				Communication error (no signal)
				Delayed communication (system will not work on time)
10	7b	System will not work as intended due to unsafe (incorrect) collision assessment signal	Catastrophic	Malfunctioning of collision probability estimator
				Incorrect input by vehicle and object status providers
				Delayed communication (system will not work on time)
11	7c	Unwanted/undesired auto braking due to false collision assessment signal	Moderate	Malfunctioning of collision probability estimator
				Malfunctioning of collision controller due to incomplete process model
12	8a	Collision with the road divider and other things, and also vehicle can slip due to missing reduce torque signal	Moderate	Malfunctioning of brake controller due to incomplete process model (incorrect brake pressure (safe brake pressure) will cause not to send reduce torque signal)
				Incorrect input by collision-assessment signal (7b)
				Communication error (no signal), delayed communication (system will not work on time)
13	9a	System dysfunction due to missing brake signal with appropriate (required) pressure	Catastrophic	Failure of brake controller components

Table 3 (continued)

No.	Step 1 no.	Hazards	Severity	Causal factors
				Brake pressure determination fails, communication error (no signal)
				Missing collision assessment signal, vehicle and object status signals
14	9b	System failure/malfunctioning as intended due to unsafe (incorrect) brake signal	Catastrophic	Incomplete controller process model
				Malfunctioning of collision controller due to incomplete process model
				Delayed communication (system will not work on time)
15	9c	Unwanted/undesired auto braking due to false braking signal	Moderate	Malfunctioning of brake controller due to incomplete process model (generation of false signal)
16	10a	System dysfunction due to missing apply brakes signal	Catastrophic	Connection broken between brake pedal and brake actuator
				Failure of braking system
				Communication error (no signal)
17	10b	False signal due to brake system malfunctioning [application of automatic brakes without need]	Moderate	Malfunctioning of brake system (generation of false signal)
18	11a	Incorrect brake pressure determination due to missing accelerator signal	Catastrophic	Sensor failure
				Communication error (no signal)
				Delayed communication (system will not work on time)
19	11b	System malfunctioning due to missing accelerator signal	Catastrophic	Malfunctioning of sensor (incorrect reading by sensor)
20	12a	Torque will not be reduced due to missing change transmission signal	Catastrophic	Component failure in the torque controller
				Missing reduce torque signal (8)
				Communication error (no signal)
				Delayed communication (system will not work on time)
21	13a	Torque will not be reduced due to missing limit air or/and fuel supply signal	Catastrophic	Component failure in the torque controller
				Malfunctioning of controller due to incorrect process model
				Missing reduce torque signal (8)
				Communication error (no signal)
				Delayed communication (system will not work on time)
22	14a	Torque will not be reduced due to missing engine switch off signal	Catastrophic	Component failure in the torque controller

Table 3 (continued)

No.	Step 1 no.	Hazards	Severity	Causal factors
				Malfunctioning of controller due to incorrect process model
				Missing reduce torque signal (8)
				Communication error (no signal)
				Delayed communication (system will not work on time)

Table 4 Mapping of the identified hazards

No.	Hazards identified by STPA	Hazards identified by FMEA
1	1a	6
2	1b	Not identified
3	2a	1 and 2
4	3a	Not identified
5	3b	3
6	4a	12
7	5a	13
8	6a	14
9	7a	Not identified
10	7b	Not identified
11	7c	Not identified
12	8a	18
13	9a	Not identified
14	9b	Not identified
15	9c	Not identified
16	10a	20
17	10b	Not identified
18	11a	8
19	11b	15
20	12a	11 and 17
21	13a	Not identified
22	14a	Not identified
23	Not identified	4
24	Not identified	5
25	Not identified	7
26	Not identified	9
27	Not identified	10
28	Not identified	16
29	Not identified	19
30	Not identified	21

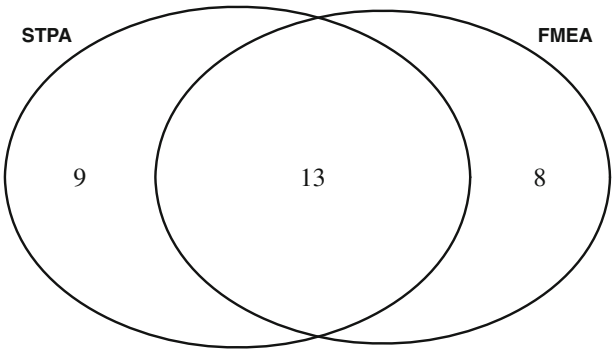

Fig. 3 Number of common and distinct hazards identified by FMEA and STPA

Each bar plot shows the percentage of the hazards that were classified in each way. For example, for hazards only identified by STPA, 18% were classified as component interaction hazards, 72% as software hazards, 54% as component failure, and 18% as system hazards. It can be noticed that the percentage of the classified hazards in the classification exceeds 100% since some hazards are classified in more than one category.

Just by looking at the "distributions" of hazard types in the three different cases, it is not possible to clearly find any major differences.

The second bar plot in Fig. 4 shows the classification of the hazards only identified by FMEA: 62% as software hazards, 75% as component failure hazards, and 25% as system hazards. As it can be noticed in Fig. 4, FMEA did not find any unique hazard of component interaction and human error type that is not identified by the STPA method. Here, one

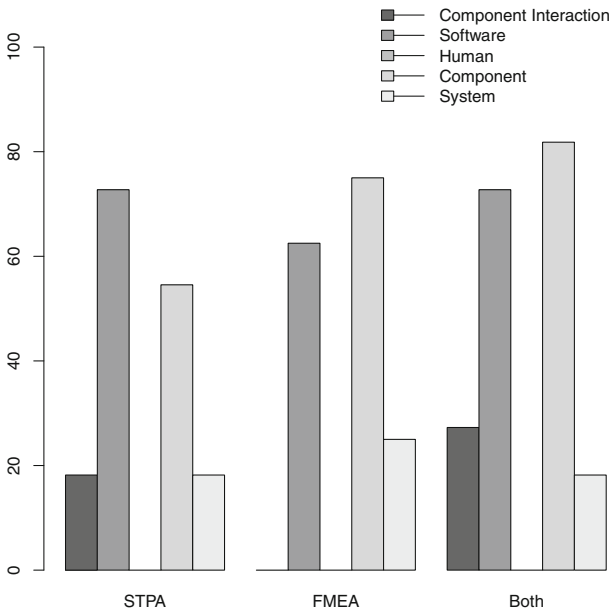

Fig. 4 Classification of the identified hazards

interesting result is that FMEA identified as many software error type hazards as STPA. It should be noted that the data points in this study are few and the focus of the study is not on quantitative comparison of the methods. However, as noted above, there is almost no difference regarding the identified software error type hazards by both methods. One positive result in favor of STPA, based on the experience of the authors of this study, is that it identified clear software error type hazards because of its keywords ("provided," "not provided," etc.), which make it simple and easy to identify software error type hazards.

Finally, the third bar plot in Fig. 4 shows classification of the common identified hazards (identified by both the FMEA and the STPA methods). Twenty-seven percent were classified as component interaction hazards, 72% as software hazards, 81% as component failure, and 18% as system hazards.

There are no common identified hazards of human error type. Apparently, none of the methods could find any human error type hazard in this study. The reason for this can be that the analyzed system does not involve much human input or interaction.

6.3 Comparison of the causal factors of the identified hazards

This section presents the comparison of the common hazards identified by both hazard analysis methods, FMEA and STPA. As shown in Table 4, 13 hazards are identified as common hazards. There is a clear difference in the identified causal factors by the two hazard analysis methods. For example, the causes identified by STPA are more detailed and cover more aspects (see Table 3). Furthermore, as one can see in Table 3, the potential causes identified by STPA cover hardware failures, communication errors including delayed communication, and software errors. However, FMEA did not find potential causes in detail. However, it found the causes that also cover hardware and software errors like "architecture erroneous or software failure" in No. 9 (see Table 1) and the potential causes are detailed enough to assess the probability of a failure.

The main reason behind the detailed identification of potential causes by STPA is the used keywords during analysis such as "provided," "not provided," and "provided unsafe" (see step 1 of STPA in Table 2). The keywords used in the STPA analysis help to identify detailed potential causes, in particular they help to find communication error causes. Based on this interpretation, it can be concluded that STPA covers more component interaction hazard causes. Here, it should be noted that FMEA also found communication error type causes, but as compared to STPA, they are not identified for all hazards and also are not detailed.

The findings of this study regarding the identification of causal factors of the identified hazards corroborate with the findings of Fleming et al. (2012, 2013) , who have compared, qualitatively, STPA with bottom-up and other top-down analysis techniques. According to Fleming et al. (2012, 2013) , STPA can find more types of causes than traditional methods, and STPA has a structured process to follow in doing the analysis that is a likely reason to result in a more complete result.

6.4 Mapping of the analysis steps of FMEA and STPA

This section presents the comparison of the analysis process of both methods. For this purpose, steps of both methods are mapped to each other in Table 5 to find the common steps. Then, the mapped common steps of both methods are compared based on the qualitative criteria derived from TAM. Here, it should be noted that the output of the mapping performed in this section is further used to compare the analysis process of both methods to yield evaluation results presented in Section 6.5.

Table 5 Mapping of the analysis steps for FMEA and STPA

FMEA	STPA	Mapping comments
Step 1: Decomposition of the system to be analyzed into subsystems and components	*Step 1*: Acquisition of functional control diagram of the system to be analyzed as a whole, and identification of some high-level system hazards to start with	*Map-A*: Step 1 of both methods are mapped as a same step in the analysis process because FMEA is based on reliability theory (decomposition required) and STPA is based on system theory (system required as a whole)
Step 2: Assigning the application function to each subcomponent and subsystem	N/A	*Map-B*: This step of FMEA does not map to any STPA step
Step 3: Determine and analyze the −potential failure modes −causes of failure −failure effects that can lead system to a hazardous state	*Step 2*: Identify the potential inadequate control commands or events (potential hazards) *Step 3*: Determine how each potential hazardous control action (potential hazards) identified in step 2 could occur (causal factors of identified potential hazards)	*Map-C*: Step 3 of FMEA is mapped to step 2 and step 3 of STPA, which consists of identification of potential failures (or hazards), their causes and effects
Step 4: Evaluate risk and calculate risk priority number (RPN)	N/A	*Map-D*: This step of FMEA does not map to any STPA step
Step 5: Specify defect avoidance or risk mitigation measures	*Step 4*: Design controls and countermeasures if they do not already exist or evaluate existing	*Map-E*: Step 5 of FMEA and step 4 of STPA are mapped to each other because they are both about designing and evaluating countermeasures

As it can be noticed in Table 5, step 1 for both methods (FMEA and STPA) is mapped as a same step called Map-A. In step 1 of FMEA, decomposition of the system is performed and on the other hand, in STPA, step 1 acquisition of the system's functional control diagram along with its safety requirements is performed. Moreover, step 1 of STPA demands for a high-level system hazards identification. We mapped step 1 of FMEA and STPA as a same step because it is an initiating step for the analysis process in both methods.

Further, step 2 of FMEA shown in Table 5 does not correspond to any step of STPA in the analysis process that is about assigning the application function to each sub-component and sub-system. Here, it is interesting to see that STPA performs this task (task of step 2 of FMEA) in its step 2 but without making it explicit. In step 2 of STPA, identification of all control commands or events of a system is performed that is more or less same as assigning the application function in step 2 of FMEA. Here, step 2 of FMEA can be mapped to identifying system's control commands or event activity in STPA but STPA's guideline does not distinguish it explicitly from other steps. In this study, the authors do not intend to modify the existing methods for sake of mapping or any other research activity instead they evaluate the analysis process of both methods based on how the methods are developed and presented along with their guidelines and application instructions.

Table 6 Assessment criteria for STPA and FMEA derived from the technology acceptance model (TAM) (Davis 1989; Davis et al. 1989)

TAM constructs	Derived qualitative criteria
Perceived ease of use	- How easy or hard
	- Why was it easy or hard
Perceived usefulness	- Provided support by the method
	- Confidence about the results
	- Applicability for software

Then, step 3 of FMEA is mapped to steps 2 and 3 of STPA (Map-C) that is about identifying hazards and their causes and effects. In FMEA, identifying hazards (failure modes) and their causes and effects is carried out in a single step (step 3). Nevertheless, in STPA, identifying hazards and their causes is carried out in two steps (steps 2 and 3).

After this, step 4 of FMEA that is about calculating risk priority number does not map to any STPA step. Here, this step gives some estimation, mostly quantitative, about the identified failure modes' severity and then based on this, their prioritization is carried out. On the other hand, STPA does not have any step that deals with identified hazards' severity and their prioritization.

Finally, step 5 of FMEA is mapped to step 4 of STPA (Map-E) that is about designing and evaluating risk mitigation measures. These steps of the analysis process that deal with countermeasures for identified hazards are exactly the same in both methods.

6.5 Evaluation of the analysis process of FMEA and STPA

In this section, the analysis processes of the FMEA and STPA methods are compared based on the technology acceptance model (TAM) (Davis 1989; Davis et al. 1989). TAM is used to investigate how users accept and use new technologies for information systems and has also successfully been applied to assess risk analysis and treatment processes (Ramler and Felderer 2015). TAM was originally proposed for information systems (IS) acceptance and usage. TAM uses "perceived usefulness" and "perceived ease of use" that estimate the beliefs in information technology acceptance and usage. The perceived usefulness means the degree to which a person believes that using a particular information system would enhance his or her job performance. Furthermore, the perceived ease of use means the degree to which a person believes that using a particular system would be free of effort (Davis 1989; Davis et al. 1989).

In this study, the criteria that we defined for evaluation of the analysis process of FMEA and STPA are derived and inspired by TAM as shown in Table 6. Here, it should be noted that the criteria were defined in a meeting after having discussion among all the authors and it is originally based on TAM.

The qualitative criteria are defined as follows:

1. *How easy or hard*: This criterion is used to assess how easy or hard it is to apply a specific step of the analysis process. For this criterion, a five-point Likert scale with values "very easy," "easy," "moderate," "hard," and "very hard" is used.
2. *Why was it easy or hard*: This criterion is used to assess if a method's particular step was easy or hard to apply then why it was easy or hard. This criterion is a follow-up criterion linked with previous criterion.

3. *Support by method*: This criterion assesses how much support is provided by method, i.e., method application guidelines and support for analysis by method itself. Here, the provided support by a method affects its effectiveness, i.e., how well a method performs in the analysis process. This support can be provided by guidelines to carry out analysis, tools to apply method, or any thing that helps analysts to carry out analysis.
4. *Confidence about the results*: This criterion is used to assess the confidence of the risk analysts about the carried out risk analysis and its results. If a used method is good enough, then the performed analysis by applying that method will yield some degree of confidence in analysts about the method application and yielded results.
5. *Applicability for software*: This final criterion used to assess how applicable a particular step of a method is for identifying software hazards and their causes. Today, almost every system has software in it that makes the system to behave dynamically. An effective analysis method must identify problems pertaining to the dynamic behavior of a system, i.e., software relevant hazards and their causes.

Tables 7 and 8 show the evaluation of the analysis process of FMEA and STPA based on the aforementioned criteria. Both methods were evaluated by using the aforementioned derived qualitative criteria.

For example, step 1 of FMEA, shown in Table 7, was easy to apply because of available detailed requirements and architecture of the system. Then, the provided support by FMEA in step 1 was the structural decomposition. After this, the analyst is confident about the results of application of step 1 of the FMEA method. Finally, step 1 was evaluated for the applicability for software, and in this case, it is well suited because it fosters risk-based development and testing.

On the other hand, step 1 of STPA was easy to apply because of the available detailed functional control diagram and safety requirements and constraints of the system. After this, the STPA method provides explicit support in this step and the analyst is confident about the analysis results. Finally, this step is well suited for software because the design of STPA is only for software that can easily be seen in the analysis. In this way, all steps of FMEA and STPA are evaluated for these qualitative criteria that are shown in Tables 7 and 8, respectively.

To summarize, the FMEA analysis process defined in Table 7 consists of five steps. Step 1 and Step 2 are easy to perform, because of the bottom-up analysis of the system. However, experience in the development of dependable systems is needed to identify failure modes (step 3) and its potential risk (step 4). Moreover, the introduction of defect prevention measures (step 5) in product development is a common task in every project. On the other hand, the STPA analysis process defined in Table 8 consists of four steps. Steps 1 and 2 are easy to apply because of the available detailed information about the system and the STPA keywords used to identify inadequate controls in the system. Moreover, step 3 is hard to perform because it finds causal factors in large amount that can be challenging sometimes. Finally, step 4 is also hard to carry out because there is no explicit support provided by the method. STPA is a simple method that does not require high level of experience by the analysts to apply the method.

7 Validity evaluation

The validity of a study represents the trustworthiness of its results, which means, for example, that the results are not biased by the researcher's own opinion or point of view (Runeson

Table 7 Analysis of the FMEA process for safety analysis

FMEA steps	How easy or hard?	Why was it easy or hard?	Provided support by the method	Confidence about the analysis results	Applicability for software
Step 1	Easy	Requirements and architecture of the system on an abstract level are well defined	Structural decomposition is supported	Experience in the application of FMEA in safety-critical systems such as railway interlocking systems was the basis	Very well suited for software, because, for instance, risk-based development and testing is fostered
Step 2	Easy	Functions of the systems are defined	Supported by templates	Method is easy to apply	Yes, on the basis of the requirements
Step 3	Moderate	The identification of failure causes may be challenging	Yes, taking domain-specific failure data into account	Confident, because a potential failure can be assigned to each task of a component	Yes, on the basis of requirements and design specification of software systems
Step 4	Hard	It is not easy to assign the potential risk to avoid a risk scenario	Yes, by assessing the complexity of a component and the probability of a failure	Taking qualitative interpretation of the RPN into account gives confidence	The method fosters the application of risk-based testing in software development
Step 5	Moderate	Efficiency of the measures have to be assessed	Yes	20 years experience in industry	The application of FMEA fosters the improvement of the software development process

et al. 2012). Validity of this kind of study (a case study) can be assessed regarding construct validity, internal validity, external validity, and reliability (Runeson et al. 2012; Yin 2003).

Construct validity considers the studied artifacts and concerns if they represent what the researchers have in mind and also if the studied artifacts are investigated according to the research questions of the study. In this study, the collision avoidance system was analyzed to identify hazards. The analysis was done by two persons, the first and the second authors of this study. The hazard analysis performed by the first author is already published in a previous paper (Sulaman et al. 2014). In the current study, the second author analyzed the system using FMEA and all the documentation and information were available, which were used during the hazard analysis performed by the first author. There could be a risk of not understanding the analyzed system and its description by the authors. To decrease this threat, a simple system was selected and also its detailed description was acquired and made available to all the authors of this study. Moreover, there could be another risk of not understanding the investigated methods by the authors. To decrease this threat, the experienced persons were selected to apply methods and also the suitable method guidelines and instructions were followed during the analyses. Also, regular meetings were carried out to eliminate any existing ambiguities in understanding of the system and its description and investigated methods. This could also impact the evaluation of the analysis process. The effect was however limited by dividing the analysis into steps that were assumed to be known and understood.

Internal validity is important and mostly applicable in studies of causal relationships. In this study, there can be a chance of *history internal validity* threat. To decrease the chance of history threat, the following measures were taken. The second author of this study was

Table 8 Analysis of the STPA process for safety analysis

STPA steps	How easy or hard?	Why was it easy or hard?	Provided support by the method	Confidence about the analysis results	Applicability for software
Step 1	Easy	The functional control diagram and requirements of system with its safety constraints are available in detail	Method does not explicitly support in this step instead it requires detailed functional control diagram and other system descriptions	Confident about the results of this step based on the reviewed literature about STPA and by studying advanced level safety course	Very well suited for software because the main focus of STPA is on dynamic behavior of systems, which covers mainly the software part
Step 2	Easy	Identification of inadequate safety controls is easy because of the STPA keywords, i.e., not provided, provided unsafe, provided too late or early, and stopped too soon	Systematic approach by using STPA keywords identified almost complete set of potential hazards	Confident because all components in system's functional control diagram are one by one evaluated against the keywords to find complete set of hazards	Very well suited as the main focus of STPA is on software and the dynamic behavior of system. It identifies majority of the hazards relevant to software
Step 3	Hard	Identification of causal factors can be challenging	Keywords to evaluate system's dynamic deviation from required safety	Confident, because STPA yielded almost a complete analysis result for both the potential hazards and their causal factors	Very well suited as it identified majority of the software relevant causal factors
Step 4	Hard	Designing new countermeasures and evaluating existing ones can be difficult or challenging	No explicit support by the method	Researcher in safety domain having 5 years of research experience in analyzing methods and tools used for the analysis of safety critical systems	It identifies problems in software and suggests improvements depending on the stage, i.e., design, development, and operation

selected to apply FMEA on the collision avoidance system. The first author already knows the existing hazards in the selected system because he has applied STPA on the selected system in the previous study (Sulaman et al. 2014). Therefore, in the current study, to improve research validity, it was decided that the first author would not apply the FMEA method on the selected system. Instead, another author did that. The second author of this study did not have access or review the previous study results (Sulaman et al. 2014). Furthermore, it was also considered that the same system information and description is available for the second author to analyze the system.

All stages of risk and hazard analysis process involve subjectivity (Redmill 2002). There is always a chance of uncertainty, the need for judgment, considerable scope for human bias, and inaccuracy. It is highly likely that the results obtained by one risk analyst are not the same to the results obtained by other risk analysts starting with the same information (Redmill 2002). In our case study, both the authors (first and second) analyzed the collision avoidance system independently by applying different hazard analysis methods (STPA and

FMEA). Moreover, both the authors have sufficient level of experience of analyzing safety critical systems and it is believed that in this case study, there is a little or no chance of this threat. Here, in this study, the objective is not to compare hazard analysis methods based on just the numbers of identified hazards; instead, the objective is to compare them based on the types of hazards found.

During the study, the results, and the written formulations of the results, were studied and discussed by all authors in order to limit the risk that results from one method were treated and formulated more positively than the results of the other. There is always the risk that other factors than the one controlled affect the result without the researchers' knowledge. We have tried to identify and remedy the most important known factors.

External validity is concerned with to what extent it is possible to generalize the findings, and to what extent the findings are of interest to people outside the investigated case. In this study, the selected system is a real software-intensive safety critical system; therefore, it is believed that the results of this study will be applicable and helpful in analysis of such type of safety critical systems. Moreover, the results of this study can be further used to compare different analysis methods using other safety critical systems. Furthermore, there might be a threat of difference in results of both hazard analyses because of different levels of experience of the first and second authors of this study who performed hazard analyses.

Reliability is concerned with to what extent the data and the analysis are dependent on the specific researchers. The reliability was addressed by conducting both the data collection and analysis as a group of researchers instead of one single researcher. In this study, there are less chances of this threat because the data used for the analysis is of third degree (Lethbridge et al. 2005), e.g., documentation, description, and published literature. Moreover, the first-degree data collected in this study is the hazard analysis results or identified hazards. To decrease the chances of reliability threats, guidelines for both methods were properly used. Special measures were taken during the hazard analysis process and continuously reviewed by the co-authors. For example, the first author of this study performed an initial comparison of the identified hazards yielded from the FMEA analysis and STPA analysis. The first author created a list of the common and unique hazards that were identified by both analysis methods. After this, the second author reviewed the initial comparison performed by the first author. The second author identified one more hazard (no. 18 in Table 4) as a common hazard identified by both analysis methods. Then, the first author classified all the identified hazards into five error categories: component interaction error, software error, human errors, component error, and system error. This classification was also reviewed by the second author of this study for the researcher triangulation.

8 Discussion

The common identified hazards are classified as software error and component error type mainly, as shown in Fig. 4. Moreover, there are some common identified hazards classified as component interaction type hazards. From the 13 common identified hazards, it can be observed that both methods found software error type hazards covering the dynamic behavior of the system. In Ishimatsu et al. (2014); Thomas and Leveson (2011); Leveson et al. (2012); Nakao et al. (2011); Fleming et al. (2012, 2013), the authors have mentioned that the traditional analysis methods (FMEA, FTA, etc.) cannot identify software errors. However, FMEA is still used in many safety critical hardware software systems and was extended to detect software hazards (McDonald et al. 2008) as well as vulnerabilities (Schmittner et al. 2014), and has even been applied for security testing (Peischl et al. 2016). Another

difference to STPA is that it does not start from an undesired state but from a malfunctioning hardware or software component. However, in our case study, there is no major difference between the types of identified hazards by both applied methods on the collision avoidance system.

Furthermore, both methods also found some hazards that are unique to them (identified by only one method, either FMEA or STPA). STPA identified 9 unique hazards that are not identified by FMEA of which majority of the identified hazards are of software and component failure type hazards. On contrary, FMEA identified 8 unique hazards that were not identified by STPA. Interestingly, the majority of the uniquely identified hazards by FMEA are also of software and component failure hazards like STPA.

Moreover, a small difference can be noticed in the unique identified hazards by the two analysis methods regarding component failure type hazards. That means, FMEA identified more component failure hazards as compared to STPA. This shows the basic philosophy behind both methods: FMEA focuses more on components, their failures, and risk mitigation measures, whereas STPA focuses on delivery of control commands and their feedbacks.

One more interesting factor in our case study is that STPA found fewer unique system error type hazards than FMEA. Because STPA is developed considering system engineering and thinking, which consider whole system as a single unit instead splitting it in several parts. One potential reason of finding few system type hazards by STPA can be that the analyzed system in this study does not have many system type hazards. On the other hand, FMEA identified 2 out of 8 hazards of system error type. Another interesting difference can be observed regarding the component interaction error type hazards; STPA identified 18% hazards of component interaction error type. On the other hand, FMEA did not identify any hazard of component interaction error type. This result corroborates with the results of the previous studies.

Fleming et al. (2012, 2013) mention that the main difference of STPA from bottom-up analysis methods like FMEA is that bottom-up analysis techniques start by identifying all possible failures. This can result in a very long list of potential failures if there are a lot of components to consider in the analysis. However, this long list is produced because FMEA takes the architecture and complexity of components into account (Stadler and Seidl 2013). Moreover, this long list of potential failures can be managed by introducing a hierarchical structure in FMEA. Furthermore, FMEA fosters propositions for the structure of a hardware and software system and generates preventive measures during development and operating (Stadler and Seidl 2013). However, in our case study, both methods were applied independently by the different authors on a collision avoidance system to find operational hazards of the system that yielded in almost the same number of identified hazards (21 by FMEA and 22 by STPA).

However, one clear difference where STPA seems to outperform FMEA is finding causal factors of identified hazards. According to Fleming et al. (2012, 2013) , STPA considers more types of hazard causes than the other traditional hazard analysis methods. Therefore, STPA is more complete than existing traditional hazard analysis methods (Fleming et al. 2012). In our case study, the results corroborate with the findings of Fleming et al. (2012, 2013) regarding STPA's complete causal factor identification.

On the other hand, FMEA (which is based on reliability theory) is stronger with respect to risk assessment of software failures by calculation of a risk priority number based on the complexity of a component or system. In STPA (which is based on system theory), there is no corresponding process step. Also assigning the application function to each sub-component and sub-system is not covered in STPA. However, the steps of (a) system

decomposition and acquisition, respectively, (b) identification of potential failures, their causes and effects, and (c) definition of countermeasures map to each other. Especially, the definition of countermeasures is according to the technology acceptance model hard to perform and requires experience.

9 Conclusions and future work

In this paper, we present a qualitative comparison of the two hazard analysis methods, failure mode and effect analysis (FMEA) and system theoretic process analysis (STPA), using case study research methodology. Both methods have been applied on the same forward collision avoidance system to compare the effectiveness of FMEA and STPA. Moreover, the analysis process of both methods is also evaluated by applying a qualitative criteria derived from technology acceptance model (TAM) (Davis 1989; Davis et al. 1989).

It can be observed that almost all types of hazards that were identified in the study were found by both methods. That is, both methods found hazards classified as component interaction, software, component failure, and system type. With regard to component failure hazards, FMEA identified more component failure hazards than STPA. With regard to software error type hazards, STPA found more hazards than FMEA of unique hazards. With regard to component interaction error type hazards, STPA found some hazards; however, FMEA did not find any of unique hazards. Finally, with regard to system type error hazards, FMEA found slightly more hazards than STPA.

Both FMEA and STPA consider system decomposition (FMEA decomposes and STPA considers whole system for analysis), identification of potential failures, their causes and effects, and definition of countermeasures. But STPA does not consider risk assessment in terms of risk priority number calculation and assignment of the application function to each sub-system.

The methods have different focuses. FMEA especially takes the architecture and complexity of components into account, whereas STPA is stronger in finding causal factors of identified hazards.

It can be concluded that, in this study, there was no type of hazard that was not found by any of the methods, which means that it is not possible to point out any significant difference in the types of hazards found. However, it can be observed that none of the methods in this study was effective enough to find all identified hazards, which means that they complemented each other well in this study.

Further research, especially in terms of case studies and experiments, is needed in order to investigate differences, but also combinations of the methods and possible extensions of them. It would be possible to carry out the same kind of evaluation and comparison by having two or more teams consisting of three to four analysts each trying out one analysis method and, in that way, investigate also the effects of teamwork in the analysis. Moreover, by having more experienced analysts for the analysis, a comprehensive list of hazards can be created that later can be used as a benchmark to evaluate results of analysis. It would also be possible to extend the analysis with more cases and explicitly add a case that has more human input to see if there are some differences in these methods or not. It would also be possible to carry out experiments to gather more data points for the analysis and, in that way, focus more on quantitative comparison than in this study.

In addition, safety has been defined as an important risk driver for testing (Felderer and Schieferdecker 2014), but the number of risk-based testing approaches taking safety analysis into account is limited (Erdogan et al. 2014). Comparing different safety analysis

methods like FMEA and STPA with respect to test planning, design, execution, and evaluation is another suggested topic for further research that could help to increase adoption of safety analysis methods for risk-based testing.

Funding information This work was funded by the Swedish Civil Contingencies Agency under a grant for PRIVAD, Program for Risk and Vulnerability Analysis Development, as well as by the Austrian Research Promotion Agency (FFG) under a grant for QE LaB-Living Models for Open Systems (FFG 822740).

References

Becker, J.C., & Flick, G. (1996). A practical approach to failure mode, effects and criticality analysis (FMECA) for computing systems. In *Proceedings of the IEEE High-Assurance Systems Engineering Workshop* (pp. 228–236).

Bond, et al. (2003). Collision mitigation by braking system. US Patent 6607255B2.

Coelingh, E., Eidehall, A., Bengtsson, M. (2010). Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC'10)* (pp. 155–160).

Davis, F.D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, *13*(3), 319–340.

Davis, F.D., Bagozzi, R.P., Warshaw, P.R. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management Science*, *35*(8), 982–1003.

Erdogan, G., Li, Y., Runde, R.K., Seehusen, F., Stølen, K. (2014). Approaches for the combined use of risk analysis and testing: a systematic literature review. *International Journal on Software Tools for Technology Transfer*, *16*(5), 627–642.

Ericson, C.A. (1999). Fault tree analysis-a history. In *Proceedings of The 17th International System Safety Conference*.

Eubanks, C.F., Kmenta, S., Ishii, K. (1997). Advanced failure modes and effects analysis using behavior modeling. In *Proceedings of DETC'97 ASME Design Engineering Technical Conferences and Design Theory and Methodology Conference Sacramento, California, USA*.

Felderer, M., & Schieferdecker, I. (2014). A taxonomy of risk-based testing. *International Journal on Software Tools for Technology Transfer*, *16*(5), 559–568.

Fleming, C.H., Spencer, M., Leveson, N.G., Wilkinson, C. (2012). Safety assurance in NextGen. Tech. rep., NASA Technical report NASA/CR-2012-217553.

Fleming, C.H., Spencer, M., Thomas, J., Leveson, N., Wilkinson, C. (2013). Safety assurance in NextGen and complex transportation systems. *Safety Science*, *55*, 173–187.

Grunske, L., Colvin, R., Winter, K. (2007). Probabilistic model-checking support for FMEA. In *Proceedings of the 4:th International Conference on the Quantitative Evaluation of Systems (QEST'07)* (pp. 119–128).

Haissig, C.M., & Brandao, R. (2012). Using TCAS surveillance to enable legacy ADS-B transponder use for in-trail procedures. In *Proceedings of the 31st IEEE/AIAA Digital Avionics Systems Conference (DASC), Williamsburg, Virginia, USA* (pp. 5D5:1–5D5:114).

I.E.C. 60812:2006 (2006). Analysis techniques for system reliability-procedure for failure mode and effects analysis (FMEA). http://www.iec.chAugust2014.

Ishimatsu, T., Leveson, N.G., Thomas, J., Katahira, M., Miyamoto, Y., Nakao, H. (2010). Modeling and hazard analysis using STPA. In *NASA 2010 IV&V Annual Workshop, NASA*.

Ishimatsu, T., Leveson, N.G., Thomas, J.P., Fleming, C.H., Katahira, M., Miyamoto, Y., Ujiie, R., Nakao, H., Hoshino, N. (2014). Hazard analysis of complex spacecraft using systems-theoretic process analysis. *Journal of Spacecraft and Rockets*, *51*(2), 509–522.

Lethbridge, T.C., Sim, S.E., Singer, J. (2005). Studying software engineers: data collection techniques for software field studies. *Empirical Software Engineering*, *10*(3), 311–341. https://doi.org/10.1007/s10664-005-1290-x.

Leveson, N.G. (2012). *Engineering a safer world: systems thinking applied to safety*. Cambridge: The MIT Press.
Leveson, N.G., Fleming, C.H., Spencer, M., Thomas, J., Wilkinson, C. (2012). Safety assessment of complex, software-intensive systems. *SAE International Journal of Aerospace*, *5*(1), 233–244.
Mäckel, M.O. (2001). Mit blick auf's risiko. software-fmea im entwicklungsprozess softwareintensiver technischer systeme. *Qualität und Zuverlässigkeit*, *1*(46), 65–68.
McDermid, J.A., Nicholson, M., Pumfrey, D.J., Fenelon, P. (1995). Experience with the application of HAZOP to computer-based systems. In *Proceedings of the 10th Annual Conference on Computer Assurance (COMPASS'95), Systems Integrity, Software Safety and Process Security* (pp. 37–48).
McDermott, R.E., Mikulak, R.J., Beauregard, M.R. (2008). The basics of FMEA. Productivity Press, paper back.
McDonald, M., Musson, R., Smith, R. (2008). *The practical guide to defect prevention*. USA: Microsoft Press.
Menkhaus, G., & Andrich, B. (2005). Metric suite directing the failure mode analysis of embedded software systems. In *ICEIS (3)* (pp. 266–273).
MIL-STD-1629A (1980). Procedures for performing a failure mode, effects and criticality analysis, U.S. Department of Defense.
Nakao, H., Katahira, M., Miyamoto, Y., Leveson, G.N. (2011). Safety guided design of crew return vehicle in concept design phase using STAMP/STPA. In *Proceedings of the 5th International Association for the Advancement of Space Safety (IAASS) Conference - A Safer Space for a Safer World* (pp. 497–501). Netherlands: Noordwijk.
Peischl, B., Felderer, M., Beer, A. (2016). Testing security requirements with non-experts: approaches and empirical investigations. In *2016 IEEE International Conference on Software quality, reliability and security (QRS)* (pp. 254–261): IEEE.
Pereira, S.J., Lee, G., Howard, J. (2006). A system-theoretic hazard analysis methodology for a non-advocate safety assessment of the ballistic missile defense system. In *Proceedings of the AIAA Missile Sciences Conference, Monterey, California*.
Pries, K.H. (1998). Failure mode & effects analysis in software development. Tech. rep., SAE Technical Paper.
Ramler, R., & Felderer, M. (2015). A process for risk-based test strategy development and its industrial evaluation. In *International Conference on Product-Focused Software Process Improvement* (pp. 355–371): Springer.
Raspotnig, C., & Opdahl, A. (2013). Comparing risk identification techniques for safety and security requirements. *Journal of Systems and Software*, *86*(4), 1124–1151.
Redmill, F. (2002). Risk analysis-a subjective process. *Engineering Management Journal*, *12*(2), 91–96.
Redmill, F., Chudleigh, M., Catmur, J. (1999). *System safety: HAZOP and software HAZOP*. Hoboken: Wiley.
RTCA/DO-312 (2008). Safety, performance, and interoperability requirements document for the in-trail procedure in oceanic airspace (ATSA-ITP) application. Tech. rep., RTCA Incorporate, Washington DC.
Runeson, P., Höst, M., Rainer, A., Regnell, B. (2012). *Case study research in software engineering: guidelines and examples*, 1st edn. Hoboken: Wiley.
Schmittner, C., Gruber, T., Puschner, P., Schoitsch, E. (2014). Security application of failure mode and effect analysis (FMEA). In *Computer safety, reliability, and security* (pp. 310–325): Springer.
Sindre, G., & Opdahl, A.L. (2005). Eliciting security requirements with misuse cases. *Requirements Engineering*, *10*(1), 34–44.
Stadler, J.J., & Seidl, N.J. (2013). Software failure modes and effects analysis. In *Reliability and Maintainability Symposium (RAMS), 2013 Proceedings-Annual* (pp. 1–5): IEEE.
Stålhane, T., & Sindre, G. (2007). A comparison of two approaches to safety analysis based on use cases. In *Conceptual Modeling - ER 2007, Springer Berlin Heidelberg, Lecture Notes in Computer Science*, (Vol. 4801 pp. 423–437).
Sulaman, S.M., Weyns, K., Höst, M. (2013). A review of research on risk analysis methods for IT systems. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering* (pp. 86–96): ACM.
Sulaman, S.M., Abbas, T., Wnuk, K., Höst, M. (2014). Hazard analysis of collision avoidance system using STPA. In *International Conference on Information Systems for Crisis Response and Management (ISCRAM)* (pp. 424–428).
Thomas, J., & Leveson, N.G. (2011). Performing hazard analysis on complex, software and human-intensive systems. In *Proceedings of the 29th ISSC Conference about System Safety, Las Vegas, Nevada*.
Yin, R.K. (2003). *Case study research: design and methods*. Thousand Oaks: SAGE Publications Ltd.
Yu, S., Yang, Q., Liu, J., Pan, M. (2011). A comparison of FMEA, AFMEA and FTA. In *Proceedings of the 9th International Conference on Reliability, Maintainability and Safety (ICRMS'11)* (pp. 954–960).

Sardar Muhammad Sulaman is a Ph.D. student in Software Engineering at the Department of Computer Science, Lund University, Sweden. He received an M.Sc. degree in Computer Systems from the Department of Computer Science, Linköping University, Sweden, in 2012. His research interests include risk analysis and management, software quality, requirements engineering, and empirical software engineering.

Armin Beer has been working in the area of test management and test automation for about 25 years. He is an independent consultant and an external contributor to the test management group of a social insurance institution in Austria. He is member of the Austrian Testing Board and of the ISTQB working party Glossary. He also lectures at the Technical Universities of Graz and Innsbruck. He has participated in presenting papers at the conferences QRS 2016, ESEM 2014, and SEAA 2014. He presented the topic "Efficient Requirements Reviewing and Test Design in Industrial Projects" at the Expo:QA'14.

Michael Felderer is a professor in Software Engineering at the Institute of Computer Science at the University of Innsbruck, Austria. He holds a Ph.D. and a habilitation degree in computer science. His research interests include software testing and software quality in general, risk management, requirements engineering, empirical software engineering, software processes, security engineering, software analytics, and improving industry-academia collaboration. He works in close collaboration with industry and transfers his research results into practice as a consultant and speaker on industrial conferences.

Martin Höst is a professor in Software Engineering at Lund University, Sweden. He received an M.Sc. degree from Lund University in 1992 and a Ph.D. degree in Software Engineering from the same university in 1999. His main research interests include software process improvement, software quality, risk analysis, and empirical software engineering. The research is mainly conducted through empirical methods such as case studies, controlled experiments, and surveys. He has published more than 70 articles in international journals and proceedings from conferences and workshops.