**Review of article "Techniques for Development of Safety-Related Software for Surgical Robots"**

Jiayin Huang 10477088

Introduction:

The research problem addressed in this paper is the development of a safe and reliable software system for surgical robots.

The paper presents a development methodology for creating a software system for surgical robots that meets the requirements of international standards and regulatory authorities. The approach involves requirements gathering, prototyping, software design, implementation, and testing, as well as the use of a coding standard and the Ada programming language.

The paper presents the development methodology used for the "EndoAssist" surgical robot, which was successfully developed and demonstrated to potential customers in a matter of days using existing electronics and low-level software. The paper emphasizes the importance of making safety the top priority and following established international standards and regulatory requirements. The use of Ada and a coding standard helped to ensure a safer and more reliable software system.

Validation and Testing:

For the development of the "EndoAssist" surgical robot software discussed in the paper, the main considerations for validating and testing include:

1. The testing and inspection procedures should focus on the system as a whole, not just the software in isolation.

2. The Test Procedure and Inspection Procedure documents should list sequences of actions and observations to confirm that the design process has produced a design that meets the requirements.

3. The Test Procedure should be followed each design change, so requirements implemented in software trace through to the Test Procedure.

4. The intended audience for the Test and Inspection Procedures is the person performing the test, who assumes a moderate familiarity with how the robot is intended to operate but no knowledge of software, electronics, or mechanics.

The author addresses the lack of experts in the field of surgical robots by emphasizing the importance of making safety the top priority and following established international standards and regulatory requirements. They propose a development methodology that involves gathering requirements, prototyping, software design, implementation, and testing, as well as using a coding standard and the Ada programming language. By following this methodology, they ensure that the software system for the surgical robot is safe, reliable, and meets the necessary standards and regulations.

Coping with Lack of Experts:

To cope with the lack of experts, the author could have considered collaborating with other professionals in related fields, such as medical professionals or experts in robotics. They could also have conducted more research and consulted with existing literature to gather information and best practices for developing software for surgical robots. Additionally, they could have sought feedback and input from potential users of the surgical robot to ensure that the software system meets their needs and is user-friendly.

The decoupled testing approach requires the software to be tested independently from the equipment it is intended to control. The software is tested in a simulated environment that is designed to replicate the expected behavior of the actual equipment. This approach enables the software to be tested rigorously and thoroughly before it is deployed to the actual equipment, reducing the risk of failure or damage to the equipment.

To achieve this, a comprehensive simulation environment should be set up that allows the software to be tested in different scenarios and edge cases. The simulation should mimic the inputs and outputs of the actual equipment, and the software should be tested under various conditions to ensure that it performs as expected.

One challenge that developers may face when developing software for robotic surgery equipment is the need for precise and accurate movements of the robotic arms. Inaccurate movements could result in serious harm to the patient.

To address this challenge, developers could implement a series of tests that focus specifically on the movement and positioning of the robotic arms. These tests could include simulations that mimic real surgical procedures, with the software controlling the movement of the robotic arms. The simulations could be designed to test a variety of scenarios, such as sudden changes in direction or unexpected obstacles in the surgical field.

Conclusion:

Additionally, developers could implement real-world testing using a prototype of the robotic surgery equipment. This would involve testing the software and the equipment in a controlled environment, such as a lab or testing facility, to ensure that the movements of the robotic arms are precise and accurate.

Overall, testing for precise and accurate movements of the robotic arms would be a critical component of software testing for robotic surgery equipment. By implementing simulations and real-world testing, developers can ensure that the software controls the robotic arms in a safe and accurate manner, reducing the risk of harm to the patient.

To test the software, we could consider model-based testing. We could also consider decision table. Decision tables make it easy to observe that all possible conditions are accounted for.

Obvious if conditions are omitted compared to traditional control structures.

Cites:

1. SSW-567: Software Testing, Quality Assurance, and Maintenance Eman AlOmar

Model-Based Testing\*

2.SSW-567: Software Testing, Quality Assurance, and Maintenance Requirements and Architecture Testing\* Eman AlOmar