# Towards Requirements for Telementoring Software

**Andrius Budrionis[a,b,e], Knut Magne Augestad[a,e], Hiten RH Patel[d,e], Johan Gustav Bellika[a,b,e]**

[a] – *Norwegian Centre for Integrated Care and Telemedicine, University Hospital of Northern Norway, Tromsø, Norway*
[b] – *Department of Computer Science, Faculty of Science and Technology, University of Tromsø, Tromsø, Norway*
[d] - *Virtual Surgical Skills Simulation Centre, Institute of Cancer, Queen Mary University of London, London, UK*
[e] - *Mobile Medical Mentor (M3) project*

**Abstract**

*The current situation in the domain of telementoring was analyzed in order to extract a basic set of guidelines and principles for building software for medical mentoring needs. The guidelines represent an initial attempt to define boundaries for the telementoring software in a globalizing health care system, where compatibility and flexibility questions come first into consideration. We employed a reverse requirements engineering technique, which resulted in definition of current guidelines for requirements looking from the implementation perspective. Moreover, interviews with potential users of the software (surgeons) enabled testing of initial requirements as well as appending some new domain specific features. The combination of reverse engineering and user interviews techniques allowed defining up to date boundaries for developing telementoring software.*

**Keywords: telementoring, guidelines, telestration.**

## Introduction

Since the first attempts to use telemedicine[1] techniques in 1960's, the domain has expanded significantly. The rapid development of information communication technologies (ICT) resulted in affordable hardware and software solutions, making the spread of technology more feasible. Growing data transmission bandwidth enabled the technology to overcome local area networks (LAN) eliminate (or minimize) distance and delay related issues. A decreasing ratio of general practice surgeons per population is reported by Williams et al. Moreover, the studies predict even higher shortage of surgeons in the future [1]. Therefore, we hypothesize, that telementoring systems could offer a direct solution to this problem, as well as promote international cooperation and education of surgeons and help to lower rapidly increasing costs of health care services [1] , [2].

As telemedicine is gaining in popularity and respect from medical personnel, new software and hardware solutions are being developed to support the increasing demand. Cases of success in using telemedicine tools are being reported [3], but Bashshur et al. raise another important question: do we have enough evaluation results to consider the cases of success as a right direction for telemedicine? To form a contribution to producing sufficient amount and quality of evaluation results two ways of assessing research in the domain of telemedicine are offered [4]:

- Testing efficacy, effectiveness and safety;
- Assessing the program according to achievement of goals and objectives **[**4**]**.

The first criterion is usually reflected in the reports of tools used, while the second one usually requires a more global outlook, which is difficult to define in a scope of a single project or usage report of a tool, therefore global assessment of goals is usually left behind. It proves the fact, that telemedicine software lacks a global direction and global requirements for the technologies under development **[**4]. Our aim is to define the fundamental principles (guidelines) for developing telementoring software using telestration[2] as a case study. Looking from software engineering perspective, the guidelines would help to understand the environment of the software without directly interfering with medical personnel. Having the background knowledge would standardize the initial requirements for the software to be developed, leaving only solution specific requirements to be defined. As the software would be built having the same basic assumptions and principles in mind, it would definitely improve compatibility and flexibility properties.

The paper is structured as follows: we firstly give a brief introduction about current problems telementoring systems are facing. As our aim is to define a set of guidelines for developing telementoring software, a methodology is represented in the next section. Moreover, the paper also gives a review of related works in the field as well as an introduction to regulatory requirements for developing software for medical needs. The paper is concluded by a set of guidelines, laying the basis for better understanding of the environment of the

[1] Telemedicine - society of American Gastrointestinal and Endoscopic Surgeons (SAGES) defines telemedicine as a set of medical practices without direct physician-patient interaction via interactive audio-video communication channel [20].

[2] Telestration – a technique, enabling drawing of freehand commands over still image or video. In this research we focus on using telestration as a tool for helping less experienced specialist perform an operation under mentorship of remote expert.

software, providing a starting point for requirements engineering process of telementoring solutions.

## Method

For extracting the guidelines a combination of reverse engineering approach and user centered design based interviews were employed.

### Reverse Engineering Approach

Reverse engineering is a common approach in redesigning legacy information systems, which usually provide critical information dealing with business processes. Sometimes it may be difficult to obtain this information directly from stakeholders, while business processes are self-evident to them and they naturally forget to mention some important details. Liu analyzed possibilities to apply reverse engineering techniques in requirements engineering process **[**5]. The author represented a model for deriving software requirements when only running software and its current stakeholders are available. The performance of the model was proven by employing it in a case study of recovering requirements for a university library system. A detailed schema of a model is depicted in Figure 1.

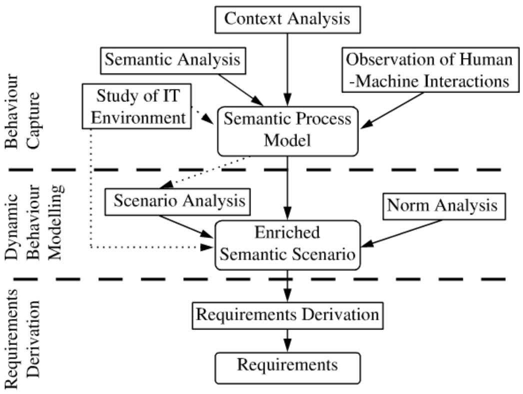


*Figure 1 - Reverse requirements engineering* **[**5**]**

The process of requirements recovery occurs in three stages: Behavior Capture, Dynamic Behavior Modeling and Requirements Derivation (see Figure 1). The first phase mainly deals with describing the environment as a main source of requirements. Moreover, a semantic analysis is also performed to gain a better understanding of the meaning of terms, concepts and functions. A basic comprehension of terms and relationships among them leads to the phase of Dynamic Behavior Modeling, concerned about modeling scenarios, rules and constraints. A representation of human computer interaction and functional requirements of the system is modeled as a Dynamic Behavior.

Looking from a modeling perspective, the first two phases yield a descriptive model. What is left, is to use the model to formalize the requirements and "validate them against the running system in a form of user acceptance testing" **[**5**]**.

Nevertheless Liu defined a part of methodology for our research [5], we were not able to fulfill it completely and reflect the outcomes in this paper. We are still lacking results from applying reverse engineering technique to fully operational telemedicine software. To compensate that, we analyzed a set of technology usage reports and papers identified in a systematic review performed by Augestad [6] instead of applying this method directly.

### User-Centered Design Approach

User-Centered Design (UCD) was selected as a basic approach for requirements elicitation from stakeholders. The main idea is to place the users of the product in the center of design process (from planning, requirements engineering to implementation and testing phases) [**7**]. We firstly aimed to identify the users of the product and define their roles in using the software. Moreover, we discussed the context (environment) of product, as a source of requirements. By interviewing the users, we defined software usage scenarios, keeping the end user in the center of the process (UCD) and minimizing possibilities of cardinal changes of requirements during software life cycle. According to the supporters of the method, active involvement of users into software process (UCD) increases the utility and usability of computer systems **[**8**]**. A basic scheme of User-Centered Design is depicted in Figure 2.

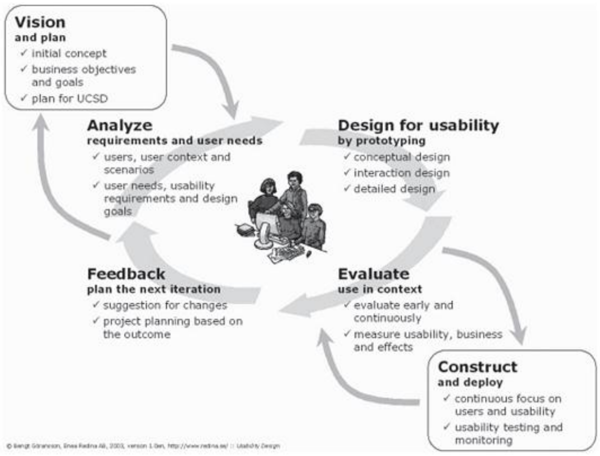


*Figure 2 - Basic scheme of User-Centered Design* **[**8**]**

Interviews were used as a main technique for requirements elicitation. Domain experts from The Norwegian Centre for Integrated Care and Telemedicine (NST) and end users (surgeons) from University Hospital of Northern Norway, Tromsø were involved in the requirements engineering process. All selected stakeholders had previous experience in using and participating in the development of tools for telemedicine. A set of interviews enabled validation of the guidelines from Reverse Engineering study and resulted in definition of ground rules for developing remote mobile telestration tools.

## Related Works

For a long time requirements for medical solutions were considered to be unique for every single implementation. But the overall globalization is making this opinion change, especially in the field of telemedicine. Therefore, we tried to identify the cases of defining the requirements in a broader sense.

The survey of related researches was performed towards finding attempts to define requirements for telementoring software. The attention was paid to telestration related requirements. The field of telestration was chosen because it combines the majority of requirements for telementoring domain, therefore guidelines for other solutions (for instance remote care) can be derived with slight additions and adjustments. The search for references was performed following the guidelines by Kitchenham [9]. After conducting the search we discovered that the majority of papers in this field are technology usage reports, mainly dealing with medical outcomes (how many patients were treated, what was the duration of the procedure, what is the ratio of success and details of the operation). Moreover, the outcomes were reflected according to different criteria, preventing generation of a global view. The most of the identified papers used similar telementoring systems resulting in overlapping sets of requirements. Therefore only the most distinctive approaches representing the most of extracted assumptions are represented in this section. As we were looking towards defining requirements in a wider scope, the reports helped identifying the most common technological solutions used for telestration. In addition, researches exploring particular software/hardware in more detail provided the contribution which was necessary towards the definition of the guidelines for telementoring software.

The closest attempt to define basic low level requirements for telemedicine software was presented by Zhang et al. **[**10]. For simplifying the process, authors aimed to distinguish the main components of the systems used for telementoring (see Figure 3). This division enabled defining requirements for every section separately. Moreover, separate sets of requirements ensured reusability properties.

As it is indicated in Figure 3, Zhang et al. focused on lower level requirements, concerning data collection, processing and transfer. Four main steps defined in Figure 3 represent the basic scheme of telemedicine solution: collection of particular data from the sensors, preprocessing, transferring to the remote location and recovering the data in an informative form for the analysis of physician. Zhang et al. were mainly concerned about modeling data transmission channel, but all other higher abstraction layer and more domain specific requirements were not represented. These requirements may look sufficient while only dealing with remote care solutions, focusing on analyzing data from a set of sensors situated in the living area of a patient. We were not able to deny the necessity to define any of the requirements discussed in **[**10**]**, but in addition to that, we replenished the set by the results from reverse engineering approach employed to analyze some well-known software and hardware in telemedicine domain, while our goal is to analyze and define the guidelines in a much broader scope.

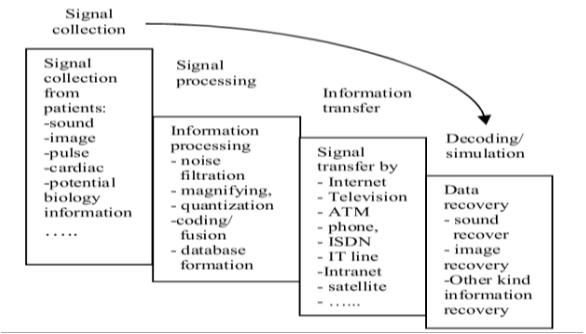


*Figure 3 - Telemedicine information systems* **[**10**]**

## An Overview of Current Solutions and Technologies

### N-Way Telestration

One of the most important recent works **[**11], introduced a new approach to visual content sharing and telestration. It contrasted traditional telestration technologies (combining video stream and markups from mentor to one stream and transmitting it to the recipient) by offering an idea of overlaying contents one over another. Conventional systems usually require dedicated hardware, limit the number of participants and file formats, geometry, distance and do not include concurrent users. Moreover, every session alters the initial video stream by adding markups made by users [11]. Therefore Zhou et al. presented a novel approach, overcoming these limits. Figure 4 illustrates the technology.

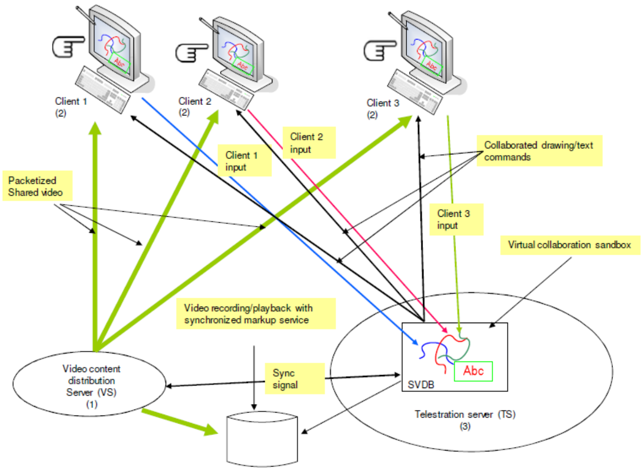


*Figure 4 - N-way telestration system prototype* **[**11**]**

The solution employs a combination of two synchronized servers: a video streaming server and a telestration server, responsible for overlaying commands on video. Client side software renders the video stream including commands coming from telestration server. Moreover, inputs from the side of the mentee[3] are also captured as telestration commands and represented on a shared video stream, making a session two- (or even n-) way interaction [11].

The main strength of the model is support for heterogeneous environments, including multi resolution screens and a variety of operational systems. These characteristics enable employing tablet computers, cellphones and PDA for developing mobile telemedicine solutions. Moreover, the solution does not pose any specific hardware requirements, making the final solution more affordable and customizable.

### Video Object Anotation, Navigation and Composition

An even more sophisticated solution was presented by Goldman et al., offering the ability to attach telestration commands to the moving objects in addition to the functionality of N-way telestration, mentioned before [12]. For

[3] Mentee – a person who is being mentored. In the domain of telemedicine it usually defines a local less experienced specialist remotely mentored by an expert.

example, symbols can be placed on the doors of a car, passing the street. The technique ensures that the symbols stay in the same position looking from the perspective of the car (see Figure 5). This functionality is a necessity while implementing telestration solutions for telemedicine, to ensure the commands are fixed to the object in a case of camera movement.


*Figure 5 - Symbols attached to a moving object* **[**12**]**

Notwithstanding some elaborate features of the used algorithms and implemented software, the main limitation comes with frame preprocessing speed and possible delays while working with live video stream. Goldman et al. tended to deal with attaching the markups to a moving object as a main goal, but expanding more on live video processing was left behind **[**12**]**.

### Surgical Robot Approach

In comparison to the techniques mentioned earlier, which can be applied not only for medical use, we analyzed a few technological examples better known to advanced surgeons. We summarize some experiences of using Da Vinci (Intuitive Surgical, Sunnyvale, CA) and RemotePresence-7 (InTouch Health, Sunnyvale, CA) robots designed to satisfy requirements of telesurgery **[**13**]**, **[**14**]**.

Both robots were employed in the context of telestration and remote presence during surgical procedures. Live video and audio conferencing services, accompanied by real-time telestration and remote control of robots were exploited. Only minor problems of both technological solutions were reported, proving the feasibility of using telementoring technique towards improving health care services **[**13], [14].

Notwithstanding the fact, that this sophisticated equipment performed as it was expected, we would like to emphasize a few characteristics, waiting to be improved. The mentioned solutions lack flexibility in employing mobile devices, therefore users are expected to have fixed positions for mentoring. Moreover, an extremely high price is also a case for consideration.

The overview exploited a set of technologies used in medical domain related to remote presence approach. The features and characteristics, regarding future works and requirements engineering for telementoring systems were highlighted. The review gave a brief introduction to the domain and some ideas for further development and improvement projects.

## Regulatory Requirements

To form a solid basis for our assumptions, standardization related documents, dealing with medical software and hardware requirements were also included as references. As no particular documents were directly reasoning with the guidelines in telemedicine domain, we aimed to analyze, extract and adopt them to support the development process of telementoring software. It resulted in contribution to the guideline derivation.

### Design Control Guidance for Medical Device Manufacturers

The regulations were introduced by the U.S. Food and Drug Administration Center for Devices and Radiological Health. The document established a framework for designing medical devices, providing flexibility for manufacturers to adapt the framework to current business processes. The main goal of techniques is to alter initial design process making systematic assessment of the design an integral part of development (see Figure 6) to ensure the deficiencies are corrected in the early phase.

Looking from the point of requirements, the framework focuses on the development process and defines some basic guidelines for every step of waterfall life cycle model depicted in Figure 6. In fact, the framework adds an extra layer of review to every step in the model, ensuring that outcome of preceding phase forms appropriate input for the next one. Extra verification of design output enables assessing if the output of design phase meets the requirements posed by inputs. Moreover, validation activities ensure that devices comply with user needs and its intended use policies. Validation should also include testing in actual (if possible) or simulated environment, reassessment of risks and possible changes. All the results of validation, verification and review processes are thoroughly documented providing a contribution to quality assurance process **[**15**]**.

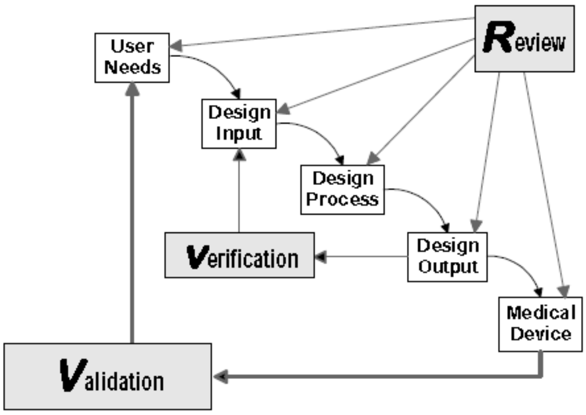


*Figure 6 - Design Control for Waterfall Design Model* [15]

### ISO IEC 62304:2006 Medical Device Software - Software Life Cycle Processes

Another of a few established regulations dealing with ensuring proper quality and reliability of medical device software was introduced by International Standardization Organization (ISO). It does not advocate any of software life cycle models, but provides a framework of processes (including activities and tasks) necessary for a safe design and maintenance of medical device software [16]. The framework can be easily adapted to supplement the software process that is being used. The standard uses criticality measure to divide final products into classes and proposes different set of practices for every class. The main idea of ensuring high quality and reliability of software is mainly based on verification procedures integrated into the software process. The standard also improves Risk Management process addressed by the International Standard ISO 14971 by adding factors related to hazards [16]. The proposed improvements of software process give detailed direction how to develop and maintain safe software systems.

## Telestration Systems. Reverse Requirements Engineering

Telestration is a critical functionality for medical mentoring applications. Moreover, as it was mentioned before, it encompasses the majority of requirements for software in telementoring domain. We see telementoring as a branch of teleconsultation discipline, providing extra functionality, adapted to ensure guidance, n-way interaction and utilization of extra features (for instance telestration) and represent immediate response. However, only a few attempts to implement and use this kind of software were identified during the review. While the majority of the papers were dealing with the transmission of still images between patient and medical personnel or representing patient’s health conditions according to a set of remote sensors, our aim was to apply a reverse engineering technique to the telestration solutions discussed earlier to extract a basic set of guidelines for developing remote presence software.

**Behavior Capture**

We analyze a behavior of the system and its environment using a schematic view, depicted in Figure 7. The initial purpose of the telestration solution is helping a less experienced specialist to perform an operation, considering that a local mentor (expert) is unavailable. Looking from the point of environment, the operating theater is an especially stressful place, posing requirements of reliability as of critical importance. On the other end, a remote expert on his portable device, dependent on possibly unreliable Internet connection, mainly concerned about the quality of video and audio to ensure accuracy. The description of the environment draws some guidelines for requirement definition which will be summarized in the next section.

**Dynamic Behavior Modelling**

After describing the environment, we continue to look at the system from the scenario perspective. As it is very specialized software, therefore only a single scenario of successful operation can be defined. Any other scenario, resulting in non-fully functional interaction has to be treated as failure, as we are dealing with a reliability-critical system.

For a schematic view, we used an operation theater equipped with 2 cameras, 2 monitors and a video streaming server. The (possibly wearable) cameras are situated in a way to create a representation of a scene as similar as possible to the one that is

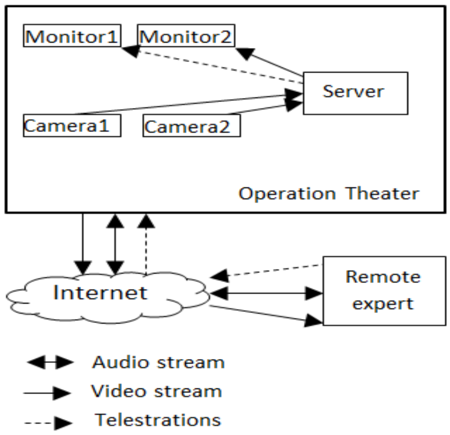


*Figure 7 - Schematic view of telestration software*

observed by the local surgeon. This ensures that the remote surgeon follows the same content as the local one. As both ends of the link are able to observe the same situation, the remote expert is able to produce telestration commands on the live video stream, which appears on the second monitor in the operation theater (see Figure 7). The telestration commands are fixed to the object in case of camera movement. Both ends of the link are able to modify the telestrations, ensuring two-way interactions, as video is not always enough to convey the current situation including exceptions.

An introduction of dynamic behavior of the system leads to emphasizing a set of requirements concerned about delay of the transmission. Moreover, usability is also a case, as we are dealing with strictly predefined procedures and we do not want the technology to influence them in an unexpected manner. As it was already mentioned above, we will summarize all the guidelines for requirements engineering in the next section.

## Results

In this section we will try to generalize all the mentioned issues towards the guidelines to be taken into consideration while designing a specific tool for telementoring. We define a guideline as a high level recommendation type input for further requirements engineering procedures. Our assumptions are based on reverse engineering approach, stakeholders’ interviews and literature review in telemedicine domain. Related Works as well as Overview of Current Solutions and Technologies sections stands for the initial source (we were not able to analyze running telementoring software) for extracting and defining guidelines for the software development process. Stakeholders’ interviews were mainly used as an evaluation mechanism minimizing the gap between reverse engineering based guidelines, extracted from the research papers, and current end users’ needs for telementoring software. We did not go into detail in discussing functional requirements in this paper, as they are dependent on a particular product and its functionality. We only lay a basis for the requirements engineering process of telementoring software.

**Flexibility**

Rapid development of technologies introduced capabilities which were not available earlier. But it also presented new challenges not only for newly developed, but also for legacy solutions. A lack of flexibility is often a case preventing applying new technologies in companion to existing infrastructure. For example, two instances of medical robots were mentioned earlier [13], **[**14**]**. Both of them were developed for stable environment configuration, which has not changed a lot from the design phase. Notwithstanding the improved data transmission bandwidth, these gadgets are not able to utilize it for better outcome. Moreover, a working position of proctoring expert is still fixed (a particular desktop computer, equipped with particular hardware) [13], **[**14**]**, nevertheless mobile technologies and devices are able to offer sufficient characteristics for a moving mentor to ensure a shorter response time in emergency cases.

While speaking about flexibility, support for different platforms should not be omitted. As we are aiming to build flexible software, ensuring high satisfaction of global goals, a platform should not become a bottleneck, limited to only predefined environment conditions. However, enhancing flexibility tends to reduce reliability, highlighting the interconnection of the properties. Therefore, Mahinda and Witworth proposed a questionnaire for evaluating the combination of flexibility and reliability (robustness) allowing to define distinct requirements that "neither overlap not contradict". The main idea is to analyze flexibility and reliability together, as the properties are in tension – "one creates change and the other resists it" [17].

It may look like we want to be prepared for every future technology change in a term of flexibility. In fact, we aimed to highlight, that flexibility in interfaces, possibility to add newly developed modules, sometimes overriding the old ones, assists adaptation to changing environment. Of course, newly developed add-ons escalate an issue of reliability requirements, which is of critical importance and will be discussed in the next section.

**Reliability**

The discussed software will be used in an environment dealing with the health of the patient. As any moment can easily change the situation from stable to critical, ensuring highest level of reliability is the first priority. The success of the procedure may be dependent on the combination of remote and local surgeon knowledge, therefore the full functionality of the software must be available at any given moment. To make the guidelines closer to the term of requirement, we offer to use a common approach of metrics for measuring reliability. Considering fully functional software as a goal, a probability of becoming not fully functional during operational period can be used as a metrics for reliability.

One of the techniques to reach a high level of reliability is to follow the regulations, discussed in earlier sections. The proposed adjustments of software process ensure a positive impact on the final quality of the software product. Moreover, continuous validation and verification activities enable easier inclusion of end users to the development process (User-Centered Design), promising higher user acceptance and meeting actual needs.

**Usability and Integration**

Operating theater is an environment having strictly predefined rules and work flows [18]. It poses a necessity of integrating software and hardware without changing established procedures significantly. An introduced solution is highly influenced by existing conditions, therefore integration guidelines must be taken into consideration in the very beginning to reach the improvement without changing workflow model. Hardware positioning can be used as an example to illustrate the situation. Operating room already contains a considerable amount of equipment, having default positions. To ensure that no changes to the workflow are made, default positions must be maintained, while newly introduced hardware is being added. Looking from the perspective of software, platform independence, mentioned in flexibility section, minimizes integration related problems, leaving hardware positioning and compatibility as the main issues.

**Risk Assessment**

Risk assessment process is an integral part of software development. It usually includes identification of potential hazards and software items that could contribute to hazardous situation, calculation of probabilities of hazardous situation to occur and severity level definition [16].

Looking from the point of mobile telementoring software, some of the hazards were already mentioned before. Dealing with mobile hardware, unreliable Internet connectivity and limited battery power can be identified as main hazards. Moreover, while performing risk assessment procedure, the possible hardware faults on the both sides of the link should be evaluated. Human factor is also responsible for creating hazardous situations while interfering with technology. Therefore, risk control measures should be established enabling evaluation of possible occurrence of hazardous situation, as well as actions that should be taken to avoid it or mitigate the consequences.

**Data Transmission**

In this section transmission delay related guidelines are discussed. It may become a problem, to ensure a considerable delay time, while one end of the link is possibly mobile, dependent on potentially unreliable Internet connection and limited battery power. These are just a few of the challenges for developing mobile telemedicine solutions, which must be taken into consideration. To summarize data transmission related guidelines, Zhang et al. approach **[**10**]**, depicted in Figure 3, can be applied. As described in earlier sections, the approach offers dividing the system into four parts and defining requirements separately (see Figure 3) to ensure separation of different categories of requirements [10]. We did not expand on this level of abstraction in data transmission, as detailed description of data transmission requirements was discussed by Zhang et al. [10]. Looking from higher level of abstraction, the main interest is towards minimizing frame transmission delay, to ensure smooth video streaming and telestrations.

## Conclusions and Future Works

Specifying requirements is one of the most important phases of software life cycle. Its significance can be proved by lots of successful and failed projects [19]. Moreover, huge costs of changes in further phases are often a result of improperly performed requirements engineering process [19].

This paper provided a basic overview of software requirements in telementoring domain. Contradicting arguments about current situation were presented, pointing out the need of changes **[4]**. As telemedicine is gaining in recognition, it is quickly spreading globally, overcoming boarders and distance. The paper escalated a problem of defining a basic set of guidelines for software and hardware products for guiding developers towards one direction. Telestration software was used as an example representing a wide spectrum of requirements in telementoring. The defined guidelines support our hypothesis that having the same initial assumption as the initial point for development process should increase the compatibility of the final solutions. The paper also introduced some new features for telementoring software. We focused on the idea of introducing mobility to medical mentoring, allowing more flexibility in working conditions and ensuring higher reachability of domain experts. In addition, by performing this research, we were also looking forward to a possibility to use more common hardware (including personal computers and mobile devices), while current telemedicine tools are usually limited to equipment by a particular vendor. Changing to regular hardware would have an impact on extremely high prices of the overall product, resulting in faster spread of technology.

We admit the weaknesses of the research, as the results were stated with no direct analysis of telemedicine software. However, the opinion of actual stakeholders, we interviewed, was highly appreciated and allowed drawing the guidelines for the research to improve telementoring software.

## References


[1] Williams, T.E.Jr., Ellison, E.C., "Population analysis predicts a future critical shortage of general surgeons," *Surgery*, vol. 144, no. 4, 2008.

[2] Wallace, P., Haines, A., Harrison, R., Barber, J., Thompson, S., Jacklin, P., Roberts, J., Lewis, L., Wainwright, P., "Joint teleconsultations (virtual outreach) versus standard outpatient appointments for patients referred by their general ractitioner for a specialist opinion: a randomised trial," *Lancet*, vol. 359, no. 9322, pp. 1961-8, 2002.

[3] Augestad, K. M., Lindsetmo, R. O., "Overcoming Distance: Video-Conferencing as a Clinical and Educational Tool Among Surgeons," *World J Surg*, vol. 33, no. 7, pp. 1356-65, 2009.

[4] Bashshur, R., Shannon, G., Sapci, H., "Telemedicine Evaluation," *Telemedicine and E-health*, vol. 11, no. 3, pp. 296-316, 2005.

[5] K. Liu, "Requirements Reengineering from Legacy Information Systems Using Semiotic Techniques," *International Journal on Communication, Information Technology and Work*, vol. 1, no. 1, pp. 38-61, 2005.

[6] Augestad, K. M., Chomutare, T., Bellika, J. G., Budrionis, A., Lindsetmo, R. O., Delaney, C. P., "Clinical and Educational Benefits of Surgical Telementoring," in *Simulation Training in Laparoscopy and Robotic Surgery*.: Springer London, 2012.

[7] Baek, E., Cagiltay, K., Boling, E., & Frick, T., "User-centered design and development.," in *Handbook of research on educational communications and technology*. New York: Routledge/Taylor & Francis Group, 2007, pp. 659-670.

[8] Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., Cajander, A., "Key principles for user-centred systems design," *Behaviour & Information Technology*, 2010.

[9] Barbara Kitchenham, "Procedures for Performing Systematic Reviews," Software Engineering Group, Department of Computer Science, Keele University , Keele, 2004.

[10] Zhang, P., Gao, B., Yiang, W., Qin, S., "Requirement Study of Telemedicine System and Product," in *2010 3rd International Conference on Biomedical Engineering and Informatics (BMEI)*, Yantai, 2010.

[11] Zhou, Q., Liu, D., "Interactive Visual Content Sharing and Telestration: A Novel Network Multimedia Service," in *14th International Conference on Intelligence in Next Generation Networks (ICIN)*, Berlin, 2010.

[12] Goldman, D., B., Gonterman C., Curless, B., Salesin, D., Seitz, S., M., "Video Object Annotation, Navigation, and Composition," in *21st annual ACM symposium on User interface software and technology*, New York, 2008.

[13] Ali, M. R., Loggins, J. P., Fuller, W. D., Miller, B. E., Hasser, C. J., Yellowlees, P., Vidovszky, T. J., Rasmussen, J. J., Pierce, J., "3-D Telestration: A Teaching Tool for Robotic Surgery," *Journal of Laparoendoscopic & Advanced Surgical Techniques*, vol. 18, no. 1, 2008.

[14] Agarwala, R., Levinsona, A. W., Allafa M., Makarova, D. V., Nasonb, A., Sua, L., "The RoboConsultant: Telementoring and Remote Presence in the Operating Room During Minimally Invasive Urologic Surgeries Using a Novel Mobile Robotic Interface," *Urology*, vol. 70, no. 5, 2007.

[15] FDA Center for Devices and Radiological Health, *Design Control Guidance for Medical Device Manufacturers*., 1997.

[16] International Standardization Organization (ISO), *Medical Device Software - Software Life Cycle Processes (IEC 62304)*. Geneva, Switzerland, 2006.

[17] Mahinda, E., Witworth, B., "Evaluating Flexibility and Reliability in Emergency Response Information Systems," in *Internation Workshop on Information Systems for Crisis Response and Management*, Brussels, 2004.

[18] Barkaoui, K., Dechambre, Ph., Hachicha, R., "Verification and Optimisation of an Operating Room Workflow," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 2002.

[19] Hofmann, H.,F., Lehner, F. , "Requirements engineering as a success factor in software projects," *IEEE Software*, vol. 18, no. 4, pp. 58-66, 2001.

[20] Society of American Gastrointestinal and Endoscopic Surgeons (SAGES). (2004) Guidelines for the Surgical Practice of Telemedicine. [Online]. http://www.sages.org/publication/id/21

[21] C. R. Doarn, "The power of video conferencing in surgical practice and education," *World J Surg*, vol. 33, no. 7, pp. 1366-7, 2009.


### Address for correspondence


Andrius Budrionis, PhD student, Department of Computer Science, Faculty of Science and Technology, University of Tromso, Norway

Email: Andrius.Budrionis@telemed.no