Ivan Evg. Ivanov[1], V. Gueorguiev[1],
D. Tcharaktchiev[2], D. Georgieva[3]
[1]Technical University Sofia, Bulgaria
[2]Medical University Sofia, Bulgaria
[3] New Bulgarian University, Sofia, Bulgaria

Medical Requirements for Selecting Local Picture Archiving and Communications Systems Influences from Information Technologies and Business Models

presenter: prof. d-r Ivan Evg. Ivanov
Technical University Sofia, Bulgaria
iei@tu-sofia.bg

Presenter

- Ivan Evgeniev Ivanov is professor at the Technical university of Sofia
- Scientific areas of interest:
  - Medical systems;
  - Telemedicine;
  - Assistive Systems;
  - Medical Imaging Processing,
  - Medical 3D imaging and live streaming;
  - e-Health
  - other areas of real-time control systems, etc.

- Memberships and awards:
  - Member of the IEEE
  - Member of IFAC TC 3.1
  - Member of Bulgarian Society of Automatics and Informatics;
  - Award of the Rector of TU Sofia (2006,2009,2020)
  - Pythagoras national science award (2020)

Introduction

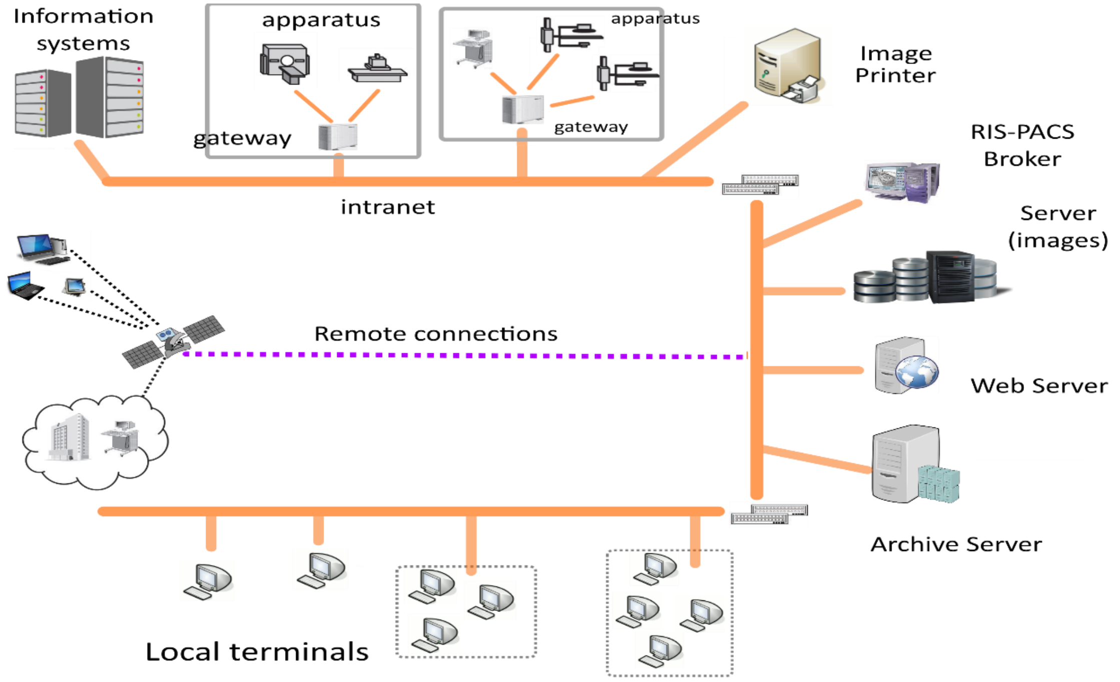
- PACS is a complete system that is responsible for the electronic storage and distribution of medical images in hospitals and medical institutions.
  - Abbreviation PACS = Picture Archiving & Communication System
  - The term PACS is now meaning “an inter-institutional computer system that governs the acquisition, transmission, storage, distribution, display, and interpretation of medical images”
- The last two decades have seen a steady increase in the clinical application of PACS.
  - This is a sustainable trend, which suggests its significant impact on the speed of changes in medical infrastructure.

Introduction

- PACS is no new approach, but some negative aspects continue to exist and limit their usability.
  - Systems are complex and expensive to acquire, replace, maintain, and repair.
  - Most often lead to increased requirements for computer and communication infrastructure.
  - An additional problem is a migration to a new version or a new type of PACS: in practice, this migration is dangerous due to the need to stop the operation of the hospital for at least one month.
- The proposed solution sounds very tempting:
  - Free "Get & Try" the system for one mount.
  - if you like how the system work buys it.

General PACS infrastructure

- PACS is a system software with a high degree of complexity and connectivity of the software solution with the hardware and communication infrastructure, necessary for the correct and full use.



The customer’s problem

- The “Free Get & Try” approach is a splendid advertising trick, but the software application isn’t an ordinary product:
  - Software is not a stand-alone product that can run on its own because the software is a set of commands that can be executed by another product (i.e., CPU-based systems).
    - i.e., buying a food, car, or home is very different from buying a new software.
  - Software is always part of a bigger system
    - We need to check not only whether it works, but also whether it does not interfere with the operation of other system’s elements.
    - The new software is installed according to the existing entities: Particularly unpleasant is the impact of software’s uninstalling on the other system’s elements.

The customer's problem

- The complexity and adequacy of the PACS selection process are a consequence of the following real business constraints:
  - The product descriptions are advertising-oriented
    - The manufacturers recommend to install the basic package for a short period and to test it in real practice but this leads to the unacceptable stopping of many medical care activities for a week, at least.
    - Functionality of the basic package is very often not indicative of functionality of the complete package.
  - Each country has specific regulatory requirements related to the storage and use of medical data.
    - Manufacturers do not provide the necessary information to assess compliance with the regulations.

The customer’s problem

- The complexity and adequacy of the PACS selection process are a consequence of the following real business constraints:
  - PACS is purchased and installed only in the long run for use without replacement
    - Manufacturers define an “ordinary” infrastructure but suggest no tools or methodology assess the difference between it and the real one.
  - The hospital's management doesn't agree with the IT department to conduct experiments with the computer-based infrastructure as these experiments potentially risk the physician's activities and the patient's health.

Our Test-based Approach

- Our approach is a task within a government-funded medical-oriented project
  - The project's board asks us to propose a unified approach to the selection of PACS, which would allow its installation in several hospitals and then connecting the installed PACS in a network.
- Like all complex systems, PACS are prone to technical problems and operational shortcomings.
  - Although it has been theoretically proven that it is impossible to find all problems by testing only, there is a clearly defined methodology that allows testing to assess the properties of a system.

Our Test-based Approach

- Key characteristics of our approach:
  - We define and apply a set of testing methods that allow us to identify those PACS that do not meet the needs of the project.
    - More than 150 types of tests are currently used in real software projects.
    - The complexity of proper testing is in the choice of the set of tests, the sequence of test scenarios, and the list of test cases.
  - We start testing with similar to the alpha testing environment testbed but we finish dynamic testing over a semi-natural model of hospital infrastructure in the final phase.
  - Instead of the classical testing, we use the ideology of risk-based testing.

How do we do it ?

- It is necessary to choose PACS from a group of software packages advertising similar characteristics.
  - To implement our approach, we use a process divided into several successive phases: Testing activities are performed first, then the results are analyzed and it is determined which products will be worked within the next phase.
- Phase #0: Preliminary phase
  - This preliminary phase helped us identify the principal goals and constraints.
    - PACS will be obtained to solve particular business tasks.
    - It is necessary to know the business goals priority within the project and for each hospital separately, as well as to know the existing business and technical limitations.

How do we do it ?

- Phase #1: Conceptual analysis
  - A static analyze was performed to check who met the list of business objectives and technical constraints.
  - After analyze:
    - The list of PACS are reduced to 40 products, which were divided into two groups: paid (14 system) and free (26 systems).
    - A list of hidden problems maybe restricts the business goals were defined respectively.
- Phase #2: Static verification by business scenario audit
  - Three groups of artifacts are used for the verification study:
    - the available product artifacts from manufacture; artifacts from projects in which this PACS has been involved; customer reviews describing problems, behavior, and undocumented characteristics.
  - After analyze:
    - lists are reduced to 4 paid and 6 free PACS.

How do we do it ?

- Phase #3: Primary dynamic testing
  - The products are installed and dynamic testing is done.
    - We use a testbed comparable to alpha testing.
  - Test cases are oriented to the following test types:
    - **Installation testing**: "install-uninstall-install-use“.
    - **Scenario-based functional testing**: predominant attention to the functions for working with images.
    - **Compatibility test**: to assess the probability of problems when PACS is starting in a real hospital environment.
    - **Interoperability test**: an accent to the characteristics and constraints of image transfer between two or more PCAS.
    - **System integration testing**: an accent to scenarios “Client - Internet - Remote server - PACS" and “Hospital workstation - PACS - DB system“
    - **API testing**: predominant attention to adding and modifying PCAS functions

How do we do it ?

- Phase #3: Primary dynamic testing
  - The products are installed and dynamic testing is done.
    - We use a testbed comparable to alpha testing.
  - Test cases are oriented to the following test types:
    - Installation testing: "install-uninstall-install-use“.
    - Scenario-based functional testing: predominant attention to the functions for working with images.
    - Compatibility test: to assess the probability of problems when PACS is starting in a real hospital environment.
    - Interoperability test: an accent to the characteristics and constraints of image transfer between two or more PCAS.
    - System integration testing: an accent to scenarios “Client - Internet - Remote server - PACS" and “Hospital workstation - PACS - DB system“
    - API testing: predominant attention to adding and modifying PCAS functions

How do we do it ?

- Phase #4: Extended dynamic testing
  - Extended testing of the principal candidate was conducted at this stage.
    - After the third phase we have principal candidate (e.g., ORTHANC for our project).
  - The testbed is a semi-natural model of the hospital's infrastructure.
    - An extended test case list to repeat the testing procedure from the third phase.
    - Usability testing, interface testing, fault tolerance testing, recovery testing, and some types of performance testing are added.
  - After analysis of testing activities
    - We confirm the correctness of the choice of the PACS.
    - We define a list of prescriptions about how to be used ORTHANC to resolve the tasks of the project.

How do we do it ?

- Final phase: Extended dynamic testing
  - Extended testing of the principal candidate was conducted at this stage.
    - After the third phase we have principal candidate (e.g., ORTHANC for our project).
  - The testbed is a semi-natural model of the hospital's infrastructure.
  - An extended test case list to repeat the testing procedure from the third phase.
  - Usability testing, interface testing, fault tolerance testing, recovery testing, and some types of performance testing are added.

Thank you for your attention !