CONCORDIA UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

SOEN 6471: ADVANCED SOFTWARE ARCHITECTURES: SECTION AA
SUMMER 2020

iCARE

The purpose of the project is to understand a non-trivial domain, to understand the needs of the stakeholders of that domain, and to become aware of the size and complexity of one class of significant software for that domain.

1. INTRODUCTION

For a number of reasons, including the recurrence of pandemics, **Health Informatics** is becoming important. There is an increasing community support for Health Informatics. For example, there are international standards for Health Informatics from **ISO**[1] and **HL7**[2], there is a book series on **Advances in Healthcare Information Systems and Administration**, and there are regular **events, such as conferences, on Mobile Health**[3].

The **World Health Organization (WHO)**[4] is an agency that is concerned with the efforts towards international public health. In Canada, efforts in the direction of Health Informatics are largely represented by **Canada Health Infoway**[5].

There is special interest in M-Health Informatics, as it is becoming a cost-effective means of identifying and monitoring health issues. For example, patients can self-monitor and transmit information to medical institutions, making M-Health Informatics particularly important to people living in remote areas, or those who are elderly or physically impaired. To medical professionals, M-Health Informatics provides access to patient data as well as access to various information resources, both of which are critical in the diagnosis and formulation of treatment. This has only been underscored due to COVID-19.

The presence of an effective **healthcare information system (HIS)** is central to the success of Health Informatics in an organization.

This project is about designing, describing, implementing a proof-of-concept, and evaluating the software architecture of an HIS, namely iCare.

[1] URL: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=54960 .
[2] URL: http://www.hl7.org/implement/standards/ .
[3] URL: https://www.google.com/search?q=Mobile+Health .
[4] URL: http://www.who.int/ .
[5] URL: http://www.infoway-inforoute.ca/ .

2. PROJECT-LEVEL CONCERNS

A team must make use of **project management body of experiential knowledge**, such as **collaboration patterns, thinklets, and articles in the book "97 Things Every Software Architect Should Know"**. For example, this can be done through practice and documenting such practice.

The work on iCare has been divided into a collection of deliverables.

DELIBERATION

A team must ensure that the work towards iCare is **visible**. It is imperative that each team **meet** regularly (at least twice a week), as well as use **Social Software**. For example, **GitHub/GitLab**, **Google Docs**, and **Wiki Hosting Service** is such social software. The selected social software should be used by a team for **communication as well as collaboration**. The **minutes of meetings** should be posted on the selected social software. The **roles and responsibilities** to be carried out by team members pertaining to each deliverable should also be posted on the selected social software. Let S(X) be social software being used by Team X. Then, the roles and responsibilities to be carried out by team members pertaining to each deliverable must be made available publicly, as early as possible (and no later than 72 hours before the submission of that deliverable).

ELUCIDATION

A team must commit to a uniform basis for terminology across its members. To do that, a **glossary** related to the project must be created and maintained throughout the project. It is preferable to develop the glossary alongside (specifically, in **parallel** with) other activities.

ABSTRACTION

A team must aim for a uniform working environment throughout the project. To do that, a **specific UML version** should be committed to, and a **single UML modeling tool** must be selected and used for all the artifacts that require the use of UML.

ATTRIBUTION

A team must strive for the highest standard of academic ethics. To do that, a team must **cite and reference appropriately** any and all non-original work (that is, any work external to that team). A comprehensive collection of resources on citing and referencing is available[6]. For example, **ACM, APA, and IEEE** provide standard formats for citing and referencing. It is important not to make claims that cannot be substantiated, and not to copy others' work verbatim regardless of whether it is cited. A copied work does not earn any credit.

[6] URL: http://library.concordia.ca/help/howto/citations.html .

3. PEOPLE-LEVEL CONCERNS

iCare is a team project involving both individual and communal work.

4. PROCESS-LEVEL CONCERNS

iCare does not follow any particular software process model per se. However, it acknowledges that large-scale software development is based on a systematic **acquisition, distillation, and communication of knowledge**. Therefore, iCare progresses through **iteration and incrementation**, follows a number of practices, is informed by experiential knowledge (such as patterns), and is driven by artifacts.

iCare must be **feasible**, by being mindful of the given time and resource constraints. To do that, iCare could be **scoped** in a number of ways. For example, iCare could focus only on specific location (say, clinic or hospital based in Montreal), on specific type of patients (say, children), on a specific medical area (say, oncology), or on specific type of mobile devices (say, mini tablets). However, scoping should not inhibit the ability (specifically, reusability and extensibility) of iCare to be used as a **product line**[7]. For example, if a distributed version of iCare has been developed, then to develop a non-distributed version of iCare should not mean repeating from scratch the design and implementation of iCare, and conversely.

5. PRODUCT-LEVEL CONCERNS

iCare must be **credible**, by avoiding misinformation and by avoiding the potential for **infodemic**.

iCare must be **accessible** within Canada by people using a mobile device. It should be possible for registered users of iCare to retrieve doctor profiles, staff schedules, and patient records in a timely manner, at any time and from anywhere. It should be possible for authorized personnel to enter patient information or modify patient information in iCare, on essentially any computing device, at any time, and from anywhere.

iCare must be **maintainable**. This is important because iCare is expected to evolve over time. For example, patient information can change.

To do that, iCare could be **inspired** by other (commercial or non-commercial, open or closed source) HIS. For example, **GNU Health**[8] and **OpenEMR**[9] are widely-used open source HIS[10]. However, cloning of an existing HIS is **not** expected.

[7] URL: http://www.sei.cmu.edu/productlines/frame_report/index.html .
[8] URL: http://health.gnu.org/ .
[9] URL: http://www.open-emr.org/ .
[10] URL: http://en.wikipedia.org/wiki/List_of_open-source_healthcare_software .

6. DELIVERABLE 1 (D1)

PART 1: VISION [20 MARKS]

This involves giving the **business goal** of iCare.

The business goal of iCare could be decomposed into sub-goals, if necessary. This could, for example, be done using the **GQM Framework**.

PART 2: PROBLEM [40 MARKS]

This involves giving the **technical definition** of iCare.

The technical definition of iCare should consist of software requirements, as outlined by the **ISO/IEC/IEEE 29148** Standard.

These software requirements must include requirements pertaining to **maintainability, privacy, security, and usability**, among others.

PART 3: STAKEHOLDERS AND CONCERNS [40 MARKS]

This involves eliciting a **mind map of stakeholders** and, based on which, giving a **list of stakeholders** of iCare, possibly organized (categorized) in some manner if deemed necessary.

It also involves giving a list of corresponding concerns, including quality attributes. (In other words, there must be a clear **stakeholder-concern mapping**.)

PART 4: VIEWPOINTS AND VIEWS [40 MARKS]

This involves giving a list of views relevant to iCare. The inclusion of each view must be **rationalized** briefly. The relationship of a view to a viewpoint model, if any, must be given.

This must conform to the **ISO/IEC/IEEE 42010** Standard.

PART 5: ARCHITECTURAL DECISIONS [40 MARKS]

This involves identifying and listing architectural knowledge (available explicitly, say, in the form of heuristics, patterns, principles, styles, and/or tactics) to be used in the software architecture of iCare. The selection of such knowledge must be **rationalized** briefly.

PART 6: PROOF-OF-CONCEPT [40 MARKS]

This involves implementing N elements, one per team member, of the software architecture of iCare, where N is the team size. The selection of such elements will be decided as the course progresses.

NOTES

Parts 1-2 should focus on the **'why'** and the **'what'**, not on the 'how'. The purpose of the **'why'** is related to **viability assessment**. The purpose of the **'what'** is related to the **problem** (namely, requirements, which may be stated informally or formally).

Parts 3-5 must conform to the **ISO/IEC/IEEE 42010** Standard, must be **informed by** each other, and be **consistent with** each other.

Part 6 must be carried out in a programming language agreed upon by the team.

In general, the diagrams, if any, pertaining to the software architecture description, must be expressed in UML, and in appropriate UML diagram types. If UML is deemed not a viable option, then the diagrams could be expressed in a "box-and-lines" form, such that each notational construct included in the diagrams is defined explicitly.

It is important that all critical arguments are **rationalized**, which may require citations to references, as necessary.

There should be a note pointing out to any **principles**[11] and/or **practices**[12] of **Agile Modeling**[13] that were deployed during conceptual modeling.

The quality of both the **representation** and the **presentation** of information will be determinants in marking. In particular, **syntactic, semantic, and pragmatic** concerns of artifacts are significant.

The deliverable must include the names of the team members and the details of contribution that are verifiable and accessible to the entire team.

[11] URL: http://www.agilemodeling.com/principles.htm .
[12] URL: http://www.agilemodeling.com/practices.htm .
[13] URL: http://www.agilemodeling.com/ .

7. DELIVERABLE 2 (D2)

PART 1: CHECKLIST FOR EVALUATION [20 MARKS]

This involves preparing a checklist of criteria for evaluation of software architecture description, a scale for satisfying each criterion, a weight for each criterion such that the weights add up to 100.

PART 2: RECIPROCAL PEER REVIEW [20 MARKS]

This involves conducting a reciprocal peer review, that is, a systematic review of D2 of another team that is based on the checklist in Part 1. It is important that the review aims to be **accurate**.

NOTES

The checklist must be based on the framework (of questions) given in the following publication: **Question Framework for Architectural Description Quality Evaluation**[14].

The criteria in the checklist must be **rationalized**.

The quality of both the **representation** and the **presentation** of information will be determinants in marking. In particular, **syntactic, semantic, and pragmatic** concerns of artifacts are significant.

The deliverable must include the names of the team members and the details of contribution that are verifiable and accessible to the entire team.

[14] URL: http://dx.doi.org/10.1007/s11219-008-9068-1 .

8. DELIVERABLE 3 (D3) [40 MARKS]

This involves, in a poster, slides, or video presentation, defending major decisions made in D1, providing a rebuttal on the comments received from D2, and reflecting on the review received from D2 (such as, by giving a perspective on the lessons learned).

NOTES

The **team in the audience** must **prepare (in real-time or otherwise) two questions and pose those questions** to the team that is presenting.