17-313: Foundations of Software Engineering

Homework 3: Requirements

In this homework, you will be eliciting requirements and documenting them for a **graduate school admission system**. Your task is to elicit requirements for this new system and document them. You will describe them using user stories and other descriptive text.

Although you already started implementing a corresponding system in the last assignment, you did not have an opportunity to gather requirements from real stakeholders. By doing so, you will see the importance of engaging project stakeholders and specifying requirements in a document rather than guessing requirements informally yourself. You may find that the requirements are substantially different from the ones you would have come up with on your own.

Learning Goals:

- Elicit functional and quality requirements from stakeholders
- Identify and describe the stakeholders and actors for a given system.
- Synthesize requirements from various sources, including stakeholder interviews.
- Create user stories documenting the functional requirements of the system.
- Develop and apply appropriate metrics to specify and measure quality attributes.

Project description (from homework 2)

CMU has over 7,000 graduate students. The university uses a software system to manage the graduate student admissions process, including collecting relevant information from applicants, accepting recommendation letters from third-parties, showing this information to the admissions committee and accepting ratings and commentary from the admissions committee to support decisions, and notifying applicants of the decision. This system is old, clunky, and universally disliked.

You and your team are developing a new system. Requirements for this new system remain a bit vague, and in fact you hope to be able to use an initial prototype to help clarify those requirements with stakeholders (in a future assignment!). In the meantime, your point of contact insists primarily that the new system needs to be "less garbage" than the current system. When pressed for details, she elaborates that faculty complain that the current system makes it too difficult to evaluate the applicants; students (applicants) and their letter writers complain that the system is hard to use. Everyone finds configuration and management difficult.

Requirements Elicitation

Your team will seek information about requirements from public documents and observations of the current and alternative systems, as well as from interviews with stakeholders.

To better inform your questions, you might want to spend some time getting to understand the domain (e.g., investigate existing systems and their documentation, including commercial competitors).

You will do two types of interviews:

- Practice interviews: Each team must do several practice interviews of a classmate who is a member of another team. We pair up the teams for you below; negotiate amongst yourselves about who on your team will serve as practice interviewees. Try to keep the practice interviewer groups to 3 or fewer students total, to keep things tractable. Your group should conduct as many practice interviews as necessary for your entire team to have participated in at least one.
- Each team must also interview 3 non-classmate stakeholders (see below). Each team member must participate in at least 2 of these interviews.

You will select stakeholders and contact them to schedule interviews over zoom yourself. We suggest you schedule 15 to 30 minute interviews with each stakeholder. The potential stakeholders listed below have already agreed to serve as interviewees for a limited number of interviews. Reach out to stakeholders early and negotiate a time for the interview. Notice that these stakeholders have real, regular jobs and may need to squeeze interviews into a busy schedule. Plan for most interviews during normal office hours. These stakeholders volunteer their time for our class and may decline interviews if they get too many requests, requests on excessively short notice, or requests that are not phrased in a polite and business-like manner.

Because of the number of students in the class, we have distributed stakeholders into groups. Please only contact stakeholders within your group. Stakeholders will decline interviews once they have reached their personal limits, time-wise, so you may need to reach out to several to find availability.

Students in non-US time zones: we realize that the stakeholders below are all located in Eastern US time. We apologize for effectively asking you to schedule your interviews in the evening, your time. If this poses insurmountable difficulties for your team, reach out to the teaching staff and we will make accommodations.

Practice interviews pairs:

- 1 - 12
- 2 - 11
- 3 - 10
- 4 - 9
- 5 - 8
- 6 - 7

Non-classmate Stakeholders:

Name	Email	Groups	Role
Michael Hilton	mhilton@cmu.edu	1--4	ISR faculty member, evaluates students applying to SE PhD and REU programs.
Claire Le Goues	clegoues@cs.cmu.edu	5--8	ISR faculty member, writes recommendation letters for applying students; evaluates students applying to SE PhD and REU programs.

Name	Email	Groups	Role
Christopher Meiklejohn	cmeiklej@cs.cmu.edu	9--12	SE PhD student, evaluates students applying to SE PhD and REU programs, a previous applicant to graduate programs.
Christian Kaestner	kaestner@cs.cmu.edu	1--4	ISR faculty member, writes recommendation letters for applying students; evaluates students applying to SE PhD program
Rohan Padhye	rohanpadhye@cmu.edu	5--8	ISR faculty member, writes recommendation letters for applying students; evaluates students applying to SE PhD program
Eunsuk Kang	eunsukk@andrew.cmu.edu	9--12	ISR faculty member, writes recommendation letters for applying students; evaluates students applying to SE PhD program
Deborah Katz	dskatz@andrew.cmu.edu	1--4	Recent SE PhD student, frequent member of the MCS admissions committee
Hanan Hibshi	hhibshi@andrew.cmu.edu	5--8	Research and Teaching Scientist, INI, evaluates students applying to INI
Tom Pope	tpope@cs.cmu.edu	9--10	ISR IT support staff who would potentially be responsible for hosting the system
Marlana Pawlak	mpawlak@andrew.cmu.edu	11--12	Staff member for MSE program who manages applications after they have been submitted.
Wode "Nimo" Ni	woden@cs.cmu.edu	1--4	SE PhD student. Member of the REU admissions committee.
Jeremy Lacomis	jlacomis@cs.cmu.edu	5--8	SE PhD student, recently applied to a number of PhD programs, helped run the REU admissions process.
Kyle Liang	kmliang@andrew.cmu.edu	1--4	SE PhD student, recently applied to a number of PhD programs.
Luke Dramko	lukedram@andrew.cmu.edu	5--8	SE PhD student, recently applied to a number of PhD programs.
Trenton Tabor	ttabor@andrew.cmu.edu	9--12	SE PhD student, recently applied to a number of PhD programs.

Proceed as follows:

- Before the interviews, become familiar with the problem, for example through content analysis of public information and other systems. Think about possible stakeholders, actors, quality attributes, user stories, and assumptions for an actual implementation. Be deliberate about who you want to interview. If you go into the interview unprepared, you are likely to miss important questions.
- Make appointments for the interviews, and plan them beforehand. You should use the interviews to check your understanding of the problem and elicit additional requirements. Since some people you are interviewing are usually not in the software field, you'll have to work with them to figure out what information is important to build the application.
- Conduct the interviews. We suggest you conduct most interviews in groups, but you may further divide roles as you prefer. Treat the interviewees and their time respectfully.
- Take notes or if (and only if) the interviewee agrees, record the interview. Microphones in most smartphones and laptops are good enough, but we suggest you test recording upfront. It can be useful to partially transcribe an interview or at least its key points immediately afterward, while the knowledge is fresh.
- Revise your understanding of the problem, based on the interview.
- Share your insights within your team and synthesize requirements, resolving potential conflicts as far as possible. If there are conflicts you cannot resolve, make reasonable assumptions and document how you would resolve them in practice.
- Complete the requirements document, filling in the specific sections indicated both within the template and below, in deliverables.
- Reflect on your experience conducting the interview and the documents in a personal statement.

Deadlines and deliverables

This homework has two (1) deadline and two (2) deliverables. The deliverables are **complete requirements document** (one (1) per team; specific sections to be completed described below) and **one reflection document** (one (1) per student). A template for the requirements document can be found on the Canvas resource page; section references in this document refer to the sections in that template.

Requirements document—Team (due Thursday, October 15, 23:59) – 120 points (75%)

Each *team* should submit a single PDF of their completed requirements document to gradescope **by Thursday, October 15, 23:59pm.**

Stakeholder and Actor Descriptions (Sections 1.3, 1.6, and 2.1, and Appendix A) - 15 points (12.5%)

In the appendix, record which stakeholders you interviewed and their roles and which team members participated in each interview.

In the main requirements document, describe all relevant stakeholders, their interests, and their importance to the system. Importance can be measured on any reasonable scale you define as long as it is clear and a developer can read the document and understand which stakeholders are key.

Describe the actors who will interact with the system as well as relevant factors about how they relate to it. Good actor descriptions give enough information about the group/individual/system that it is easy to identify their motivation in interacting with the system; their primary goals/needs in doing so; how they typically interact with the system and their level of skill in doing so (when referring to humans); and other system-relevant factors.

Functional requirements (user stories; Sections 2.2 and 2.3) - 40 points (33%)

Document the functional requirements in the form of *user stories*. Provide at least two user stories per student in your group.

For each user story, you should also include a discussion of two questions.

1) How will you test this user story? If you cannot test your user story, it is overly vague. This is also a signal that you don't really understand it yet. For every user story you tackle, you should be able to describe not only the user story, but how you will test it.

2) When is this user story done? This is related, but a slightly different question. Many teams will include a "definition of done" when developing user stories. This can help prevent misunderstandings when it comes to who is responsible for work such as implementation and more.

3) An estimate as to how long this user story will take to implement.

Additionally, you will need to prioritize the requests you get from your stakeholders and decide which of them you plan to support in your system (see also conflicts below). You should not write user stories for everything your stakeholders mention; you should choose wisely according to a set of features that is cohesive and represents a reasonable first iteration of a product.

Quality requirements (Section 3) 25 points (21%)

Define and order the *five* most important quality attributes within the context of the system. A good description of quality attributes unambiguously defines what the requirements mean within the context of the system; why each requirement deserves the given rank; how to measure each requirement; and how those metrics should be interpreted. If two or more quality attributes are equally important, it is acceptable to give them the same rank, but be sure to justify why they are equally important.

Conflicts (Appendix B) 15 points (12.5%)

Briefly describe and discuss all conflicts you identify during elicitation. For conflicts that you could resolve, describe how you resolved them. For conflicts that you could not resolve within this assignment, briefly outline how you would resolve them if you were doing this for a real customer (e.g., be concrete of whom you might ask which question and how answers would inform your decision).

Quality Assurance (Appendix C), 25 points (21 %)

In class, we discussed the INVEST method for quality assurance of user stories. For N user stories, where N is the number of people in your team, explicitly argue why the user stories do follow the INVEST criteria. If you have to make tradeoffs in one dimensions because of another dimension, you should argue that tradeoff explicitly.

Reflection document – Individual (due Thursday, October 15, 23:59) - 40 points (25%)

In a reflection document of two pages or less (soft limit) describe what insights (if any) you gained from going through the requirements process, including conducting interviews and synthesizing and documenting requirements. The reflection can include both positive and negative aspects and can set them in context with experience outside of class, if applicable. Use specific examples from your experience gained in this assignment, and include any context information required to explain the situation.

The following questions may help you to identify interesting issues for the reflection document, but it is not necessary to answer them all:

- Were there surprising aspects of the requirements process?
- What went well in the preparation for and during the interviews? In retrospect, what could you have improved?
- What did you not expect in the interviews? Were there unusual situations or misunderstandings? Were there unexpected requirements that you did not anticipate?
- Who else would you interview, and why, if you had access to more stakeholders or more time?
- Did the interview and requirements document cause you to adjust your initial assumptions in any way? How so?
- Were there surprising conflicts in the requirements? Were they easy to resolve?
- How do the current system requirements compare to the (implicit) requirements used to create the feature your team developed in Homework 2? What would you do different for Homework 2 with that information?
- How do current systems requirements fit with the decision to build on the Mayan EDMS platform? Are there any requirements that are less feasible given that you are building on the Mayan platform?

Logistics

Schedule your interviews as soon as possible. Please contact your TA if you have any issues. You will need to be respectful of the fact that the stakeholders are real people with their own schedules and responsibilities; if you wait too long, you may not be able to complete your interviews.

Grading

You will be graded as a team, with an individual component. This homework is worth 160 points, as divided above. We provide additional detail on what constitutes full-credit solutions in the narrative above; a summary is below.

For the group component, we expect all below-listed information to be appropriately organized in a document that follows the template we provide. Additionally:

To receive full credit on the interviews and stakeholders and actor descriptions, we expect, based on your interviews:

- A concise indication of whom you interviewed and their role in the system. For practice interviews, list who was interviewed (and who participated in each) and indicate the role the interviewee pretended to play.
- A complete and well-described set of relevant system stakeholders, including their roles, interests, and importance in the system.

- A complete and well-described set of actors, including their role in and, mode of interaction with the system, and other system-relevant features.

To receive full credit on the functional requirements, we expect:

- At least 2*n* user stories (for a team with *n* members) that describe important system functionality at an adequate level of detail and quality.
- For each user story, in addition to the "as a _____..." we expect the following additional info:
    - How you can test this user story
    - When the user story will be done
    - How long you estimate it will take to implement this user story (with a brief justification for the estimate)

To receive full credit on the quality attributes, we expect:

- The identification and description of at least five (5) quality attributes, including a ranking of their importance and a coherent justification of that ranking.

To receive full credit on the conflicting requirements, we expect:

- A description of all conflicts you identify during elicitation.
- For conflicts that you can resolve, describe how you resolved them. For conflicts that you cannot resolve within this assignment, briefly outline how you could resolve them if you were doing this for a real customer

To receive full credit on the quality assurance discussion, we expect:

- an explicit discussion of why N of your stories follow the INVEST principles, with tradeoffs justified.

To receive full credit on the individual reflection, we expect:

- A detailed, well written, and well structured reflection on the requirements elicitation, interview, and synthesis and expression process.
- An analysis beyond mere descriptions and superficial statements, including supporting evidence for claims, that reflects on the causes of identified issues or your own experience.