Project Plan Instructions

"By failing to prepare, you are preparing to fail." – Benjamin Franklin

Goal: Demonstrate that your team has a thorough understanding of

- what features the customer wants,
- what the user interfaces will look like,
- what architecture/design the system will have,
- how your database will be organized (if your system will use one),
- how you will assure quality of your product,
- how you will manage the project,
- what special resources you will need,
- what schedule the work will follow (best guess), and
- what risks your team must address.

Deliverables: Plan artifacts (for plan-review meeting) and presentation.

Note: Any artifacts produced may created in PPT form or as Google documents, wiki/web pages, etc.

Plan Components

Requirements

To facilitate communication among your customer, manager, and team, your team must create a **list of requirements** that describes the system's functionality. Follow these criteria:

- Your requirements may be documented in the form of user stories, a structured list of natural language specifications, or use cases.
- Requirements should be documented at a level of granularity appropriate for dividing up and managing the work (e.g., they should be reasonably estimable).
- You must elicit the requirements from your customer. This will likely require some back-and-forth communications with your customer.
- Your customer must approve the set of requirements in your team's plan (prior to the plan-review meeting and ASAP so you can move forward with the other planning).
- All requirements should be understandable to your customers (no technical/implementation details).
- For your initial plan, you must make the set of requirements as complete as possible (understanding that they will probably change as the project moves along).
- Your team must maintain this set of requirements throughout the project, keeping them up to date.

User Interface Design

Your team must create rough sketches of what the various system interfaces will look like.

- Lo-fi renderings are preferred at this stage. For example, hand-drawn pictures with notes are fine. Sketches done in, say, PowerPoint would also be acceptable.
- Your customer must review and approve your interface designs (prior to the plan-review meeting).
- Your sketches should be as complete as possible (include all key features, interaction dynamics, effects of button clicks).
- Designs may include different levels of refinement (e.g., "basic" and "advanced" versions).

Software Architecture/Design

Your team must describe what software components you will implement and how those components will interact.

- You have some flexibility in how you communicate this. For example, data flow and/or deployment diagrams are probably appropriate.
- Make it clear where the users fit in.
- Specify all hardware/software technologies to be used in implementing your system.
- Specify all programming languages and/or platforms (e.g., Java EE) that you will use, and how they fit into your design.
- Keep your descriptions somewhat high level; for example, I'm not looking for detailed class diagrams.

Data Model

If your system will use a database, create a database design (i.e., a detailed data model of the database).

- You may represent the model as an object-oriented class diagram, an entity-relationship diagram, or other common database modeling notation.
- Make your tables/design as complete as possible. For example, if you will use a relational database, such as MySQL, design all the tables, keys, etc.

Quality Assurance

Your team must define what policies and procedures will be followed and what technologies will be used to ensure that the software built is of good quality—especially from the perspective of the customer.

- A plan for testing must be included (both unit and system testing).
- Be clear about which verification/validation provisions are automated versus manual.
- Describe any specific tools to be used for verification/validation.
- If part of the plan involves the customer, make sure that the customer is OK with it (prior to plan-review meeting).

Project Management

Your team must define what policies and procedures will be followed and what technologies will be used for software configuration management, collaborative development of both code and other artifacts, and bug/issue tracking.

- Be sure to address version control and communication.

Resources Needed

List any special resources required beyond your individual desktop/laptop computers. Specify how you will get the resource.

- Address resources related to hosting the source repository, bug tracking, system testing/integration machines, and any others particular to your project.
- If you expect the customer to provide a resource, make sure that they agree.

Schedule

List the expected features to be completed during each iteration.

Risks

As your team plans the project, you will find that there are key unknowns, things may not work as planned, or things that could just plain go wrong. These are *risks* to the success of the project, and your team must identify them early.

- Classify the difficulty of each risk: for example, “no idea how to do” versus “probably doable.”
- Classify the importance of each risk: for example, “showstopper” versus “nice to have, but not vital.” Focus on the most important risks.
- If a risk can be eliminated by talking to customer, do so ASAP (prior to the plan-review meeting).

Submission

For your submissions, create a folder in either Google Drive (https://www.google.com/drive/), Dropbox (https://www.dropbox.com/), or OneDrive (accessible through Memphis mail).

Upload your files and share the folder with your team, instructor (sdflming@memphis.edu), and GAs (mluong@memphis.edu, tbaniya@memphis.edu). Only one shared folder per team.

Checklist for Plan-Review Meeting

- ❏ List of requirements created as completely as possible.
    - ❏ Requirements approved by the customer.
- ❏ Sketches of user interface design.
    - ❏ Designs approved by the customer.
    - ❏ Label key features, buttons, etc.
- ❏ High-level diagrams of software architecture/design.
    - ❏ Identify programming languages and/or platforms you will be using.
    - ❏ Identify hardware/software technologies used to implement your system.
- ❏ Detailed design of database (if database is used in system).
- ❏ Plan of policies, procedures, and technologies used to ensure good software quality.
    - ❏ Customer approval if needed in testing.
- ❏ List of policies, procedures, and technologies used for project management.
- ❏ List of special resources (and how you will get it).
- ❏ Plans of expected features to be completed for each iteration.
- ❏ List of risks in this project.
    - ❏ Identify difficulty of risks
    - ❏ Identify importance of each risk