Project Overview

For this project, we created two to three web-based games. The purpose of these games was to help cut down time spent in meetings by allowing players to provide and receive background information about the meeting on their own time. The three planned games were Collaborative Cover Story, Anonymous Brainstorming, and Empathy Map. These games were built upon a framework that supported users and provided incentives by rewarding users who participate in the games. Any user registered with the system would be able to create a game in which they would assume the facilitator role. Once the game had been created, the user could decide which participants should play, likely the ones that would be attending the meeting. Once the game was setup, the users invited to the game would be notified and could begin playing based on the rules set by the facilitator. After the game had finished, an artifact would be produced representing the end state of the game and sent to the facilitator and/or the participants.

The Collaborative Cover Story game was a turn-based game in which the facilitator creates a headline. The players would then take turns writing one sentence each to form a story about the headline. Players would be able to upvote or downvote any sentences they like or dislike. Sentences would be emphasized accordingly. The final artifact of the Collaborative Cover Story will ideally look like a newspaper, with the headline at the top and the emphasized sentences looking like newspaper articles.

The Anonymous Brainstorming game would be an asynchronous game in which the facilitator would provide a topic and players would contribute ideas anonymously. Players can “Like” sentences, which will not have any effect until the game is over. The final artifact of the Anonymous Brainstorming will be a list with all of the ideas sorted by number of likes.

The Empathy Map game would be a brainstorming game in which a picture, likely of a customer, would be placed in the middle of the board. The area surrounding the picture would be split into sections, each section representing one of the customer’s senses (Sight, Sound, Feel,
etc...). Players would be able to place ideas about what the customer sees when using this product, or how they feel, or what they hear, etc... Players would also be able to vote and the final artifact would reflect the number of votes of the ideas in each section.

The purpose of voting in all of the games is tied into the gamification concept at the heart of the system. When a participant does well they receive votes which translate to points. These points allow them to level up and collect achievements which creates a reward system similar to video games. Inclusion of this system allows for MITRE to test the value of gamification in a corporate setting.

The system was not intended to be a lasting product. The intent of the product was to be a proof of concept for gamification techniques with these pre-meeting activities. If this proof of concept works, the system will be used as the centerpiece of a presentation to create a budget for developing a new system, adhering to the sponsors standards.

**Basic Requirements**

The Framework:

The Mitre Gamified Facilitation System shall allow users to login to this web-based system and view a dashboard that shows any games that they are participating in. Users shall then be allowed to select any game, from this dashboard, that they are part of and view the game. Users will also be able to create games from the dashboard. The game creation will take users through a wizard of selecting all of the game options and adding users to play the game. At any time, the users may return to the dashboard. Also on the dashboard, users will be able to view score-related metrics that they earn playing games: total score, user level, any awards or achievements, as well as a leaderboard of users in the system. Users will also be able to change their account settings and log out at any time.

Cover Story:

For the Cover Story game, users will take turns writing one sentence of a story based on a headline set by the game creator. Users will be able to upvote and downvote sentences in the story to increase or decrease how much they stand out. Voting also affects the score of the user that wrote the sentence. The system will be able to produce an artifact representing the current state of the system at any time. The artifact will show the headline, along with all sentences highlighted appropriately so as to appear like a newspaper.

Game Creation Options:

- Headline
- Turn Length
- Number of up/down votes
- How to emphasize sentences (Boldness/Size/Opacity)
- Whether or not to show the list of users and/or turn order.

Anonymous Brainstorming:
For the Anonymous Brainstorming game, users will be able to submit any ideas that they have based on a topic set by the game creator. This game will accept any ideas from any player at any time. All ideas will be displayed, most recent first, in a list with no indication of the submitter. Users will be able to upvote any ideas at any time. Upvotes to a users idea increases their score. The system will be able to produce an artifact representing the state of the game upon completion. The artifact will show all of the ideas submitted in order of number of votes received.

Game Creation Options:
- Brainstorming topic
- Number of upvotes

Empathy Map:

For the Empathy Map game, users will be able to submit ideas for any of the categories at any time. The ideas will all be viewable within their respective sections with no indication of their submitter. Users will be able to upvote or downvote ideas at any time. Voting will affect the score of the submitter of the sentence. The final artifact will reflect the number of votes each idea received.

Game Creation Options:
- Center Person’s Name/Description
- Section titles
- Number of up/downvotes

Constraints

The majority of the constraints of the project were related to the class itself. This limited the time to two academic quarters to complete the project. This also limited the personnel to four students whom all worked for no compensation. The few constraints on the project specified by the sponsors were that the project must be web-based, the system must be usable on a mobile device, and the system must be deployable to a windows server. The sponsors specifically mentioned that the technology, methodology, and everything else related to the project was up to the discretion of the development team. This was to remove any red tape.

Being usable on a mobile device was clarified to mean that native mobile apps were not necessary, a responsive web-app would be sufficient.

Development Process
We used a modified version of Scrum. Our modifications were based on our team size and the distributed nature of the project. We had daily standup meetings over Google+ with the development team and weekly meetings over Google+, then Lync, with the project sponsor. Our Scrum Master, who was the main point of contact for the sponsor, was also a developer due to the size of the team.

Our process was approved by our sponsor and proved to be an appropriate choice as our sponsor frequently changed/re-evaluated the requirements and asked multiple times for faster turn-around. We used 2 week development sprints, sending a potentially shippable product to the sponsor for acceptance testing.

Each member of our team did development work, but as the project progressed, one member became dedicated to user interface work and the Scrum Master dealt mostly with customer interaction and deliverables.

Project Schedule: Planned and Actual

The team developed the project schedule based on our development methodology of using Scrum with two week sprints. The divided nature of the parts of the project also played a role in developing the schedule. The main milestones that were included in the schedule were:

- Basic framework with the dashboard
- Cover Story
- Anonymous Brainstorming
- Code freeze

The team stuck to the schedule very closely, the only times when the planned and actual schedules did not align were during requirement changes from the sponsor. At these times, the team took the new/changed requirements into account and altered the planned schedule for everything to fit in the same timeline.

System Design

GWT...
Client-server...
Client side...
Server side...
Talk about how we started with a concept of a meeting, then had to refactor/redesign the system to work around getting rid of that...we can talk about how we used a work-
around(keeping meetings in the back-end) since the system will be thrown away, but if it were going to stay, we would have renamed classes and refactored a bit more...

### Process and Product Metrics

The three metrics that were tracked were Velocity, Estimation Accuracy, and Time. Velocity, which was an obvious choice as we were using Scrum, showed us how much we were getting done in each sprint. This allowed us to better prepare for the upcoming sprint. We averaged between 40-50 story points per sprint, when planning out sprints, we generally overestimated with the expectation that we may not get to everything. This allowed us to estimate these extra tasks as a team. For example, if we were near the end of a sprint and I had finished my current task, I would take the next task off of the sprint backlog, but if the sprint backlog was empty, I would take an unestimated task from the product backlog. Having the extra one or two tasks already estimated eliminated this. By tracking our velocity, we had a good idea of what we could get done in a sprint and we could plan ahead a little based on that to give product milestones.

Our Estimation Accuracy metric allowed us to track how well we were estimating the amount of time and effort tasks would take.

Tracking our time allowed us to see at the end of each sprint what we spent the most time on. This metric helped us to judge our estimation accuracy and provide more of a development effort to the tasks that were taking more time.

### Product State at Time of Delivery
At the time of delivery, the team produced two of the original three games, as agreed upon by the sponsor as the project progressed, along with the framework which they were built on. Both of the games were accepted as feature complete by the sponsor prior to the delivery of the system. The sponsor was still undergoing usability testing and gathering actual data from using the system to use toward their sales pitch.

The framework of the delivered system allowed users to create/manage accounts, create games, view any games they were participating in, and view their score/level. Although the achievements have not yet been implemented in the system, there is a placeholder for them on the dashboard.

All discrepancies between the final delivered product and what was originally planned at the beginning of the project were based on requirement changes by the sponsor and were negotiated between the team and the sponsors.

The delivered product is in a good state for its intended use as a sales pitch. Any further development that is needed before the sponsor demonstrates the product will be done by sponsor co-ops. If the demo is successful, the project, as it is now, will be scrapped and development will start from the beginning adhering to the sponsor’s development standards.

**Project Reflection**

Overall we feel that this project went very well. Our choice of technologies and methodology meshed nicely with the demands of the sponsors. They wanted to have high visibility into what we were doing and to have quick, testable prototypes. Our choice of scrum allowed us to do that. As the project progressed, the sponsors wanted faster and faster turnaround for features and bug fixes, at this point we started using continuous integration of our code. This allowed the sponsors to immediately test any code updates we had. This is the one thing that we really wish we had changed, we wish we had set up continuous integration from the beginning. We realized that no project is too small for it. In terms of technologies we really have no regrets. Knowing from the beginning that we were developing a sales pitch and that everything would be thrown out, we made all of our technology decisions on quicker things that we were more familiar with. If our code was going to be put into production, we may have made some different choices, but since that is not the case, we are happy with what we delivered.