PROTOTYPE

CS577a Fall 2014
WE ARE TROJANS (WAT) NETWORK
Team #1

Pittawat Parnchaisirikij (Nam)  Punyawee Pakdiying (Golf)
Kamonphop Srisopha (Title)    Eirik Skogstad
Suleyman Erten               Min Li (Deni)
Saloni Priya                  Ameer Elkordy
AGENDA

● Evolutionary Prototyping via UI Designing

● An algorithmic approach to solve an issue relating to the core capability
1st Risk Identification - User Interface

Evolutionary Prototyping via UI Designing:

- **Why?**
  - No concrete requirements of the UI
  - Need feedback from the clients
  - Need to enhance the team’s understanding of the UI

- **How to mitigate the risk?**
  - Using *Buying information* methodology
  - Demonstrate the UI visually
  - Get feedback from the client
Client Requirements:
(UI should have)
- Event System/News Feed
- Leaderboard
- Notification System
- Profile
- Redemption
- Search Option
Client Requirements:
(UI should have)
- Categorization
- Q&A Forum
- Like Functionality
- Dislike Functionality
- Search Option
Client Requirements:
(UI should have)
- Editing Profile Information
- Editing Profile Photo
- Changing Password
- Recent Activities
- Current Points
2nd Risk Identification - WAT Points System

An algorithmic approach to solve an issue relating to core capability:

Why?
- Crucial feature of the system
- Determine feasibility of the proposed points system
- Create mutual understanding among team members

How to Mitigate this Risk?
- Formalize rules of the points system
- Using a Flow Chart Diagram
- Get feedback from the client
Points system (WAT Points) overview

Each user has 3 different points

- **Total points (TPoint)**
  - Accumulated lifetime points of a user
  - *Purpose of this point*: Identify the credibility of the user

- **Semester points (SPoint)**
  - Reset every semester
  - *Purpose of this point*: Compete with other user in the system

- **Current points (CPoint) or Usable points**
  - Redeem items from gifts store.
Work-Breakdown Structure Specific to “WAT” Points Functionality

* But for gain and lose, it can be different for different point.
Current Issues

● Requirement: A post can only positively contribute to user points
  ○ 2 likes, 1 dislike = (2 * 2) - 1 = 3
  ○ 1 like, 3 dislikes = 2 - 3 = -1 but user gets 0 points

● Usable point balance can become negative

@Copyright 2014
How to calculate Post’s points

Requirement: A post can only contribute positively to a user’s points.

NL = Number of Likes
ND = Number of Dislike
LV = Like Value
DV = Dislike Value

\[ Post’s\ Point = \begin{cases} 
0 & \text{if } NL.LV - ND.DV < 0 \\
NL.LV - ND.DV & \text{if } NL.LV - ND.DV \geq 0 
\end{cases} \]
Algorithmic Flow-Chart:

1. Dislike a post/comment
2. Like a post/comment
3. Keep the Number of Like and Dislike
4. Calculate the old post/comment's points
5. Increase Number of Like/Dislike of the post/comment
6. Calculate the new post/comment's points
7. New post/comment points is greater than the old points
   - Yes: Increase the poster points equal to the point's different
   - No: Reduce the poster points equal to the point's different

@Copyright 2014
“Usable points” issue

- Use negative points balance?
- Monthly points update?
We Are Ready for you Questions..
Thank You for your Time..