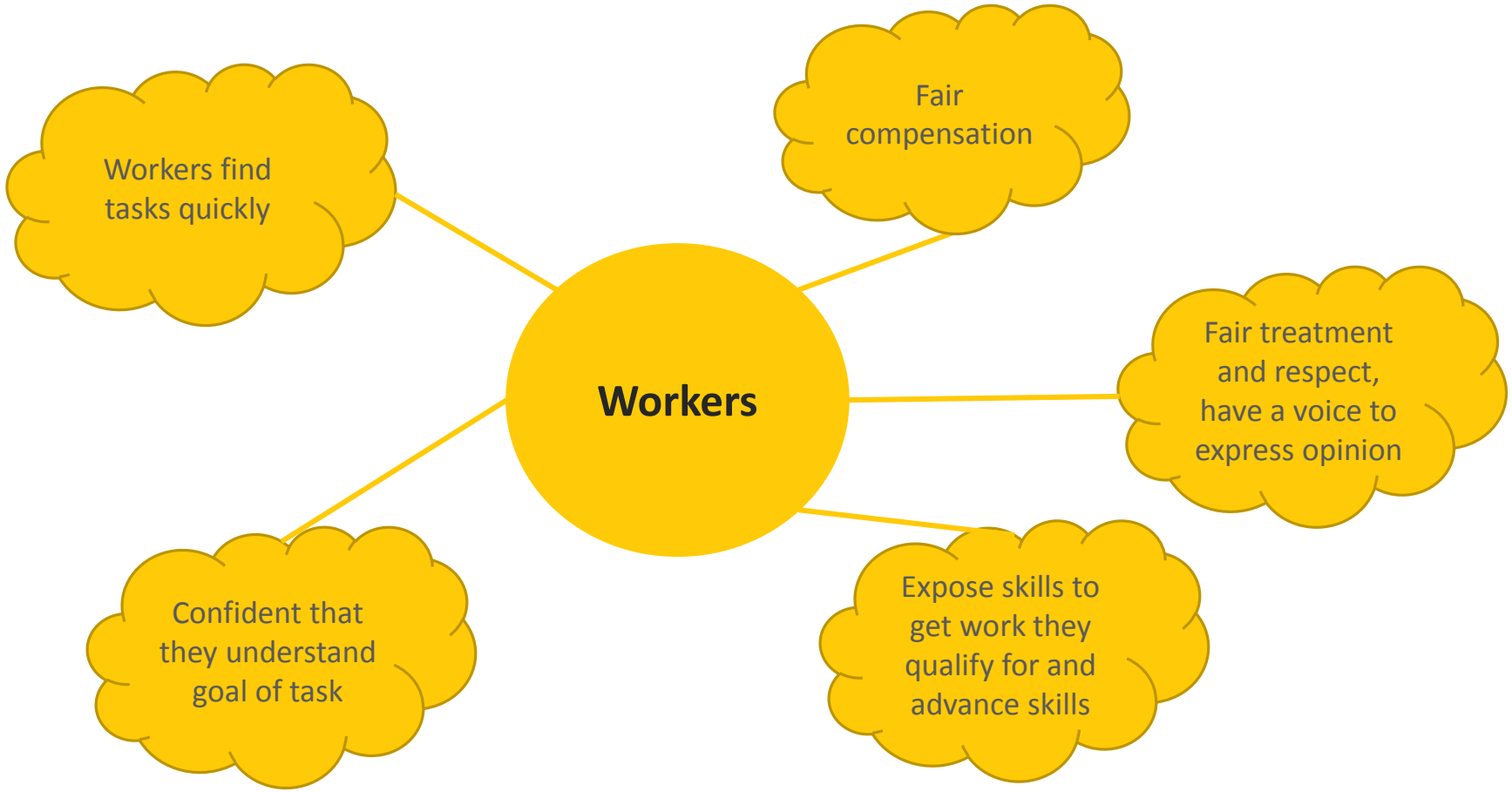
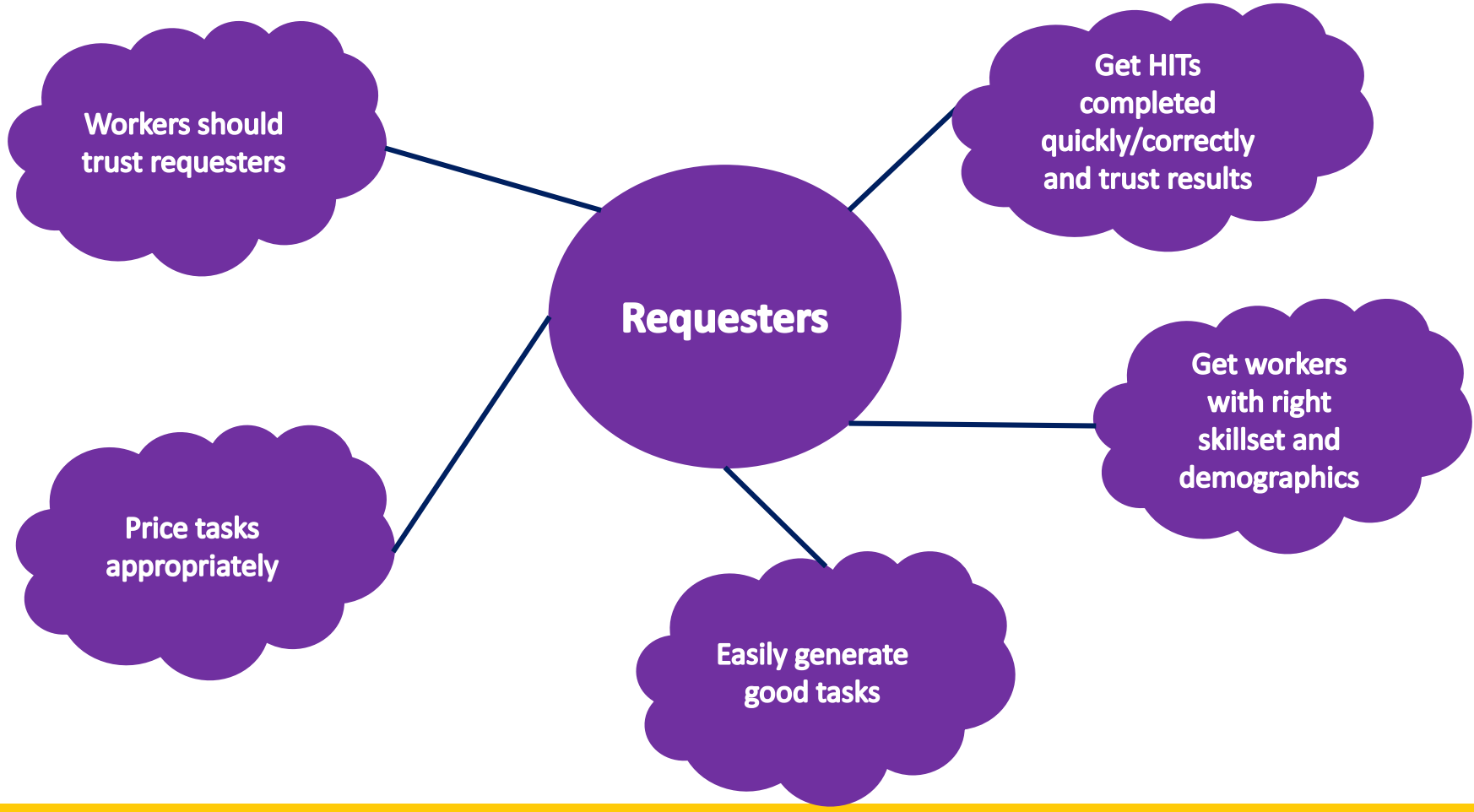
Milestone 3

TEAM PIXEL PERFECT

Worker Need Cloud



Requester Need Cloud



New Rating System

- We draw inspiration from the cult game *Need For Speed : Hot Pursuit*, where players can choose to play either as a racer or a cop. Players level up in their respective careers and earn rewards and tools to help with their racing and chasing. Players earn XP based for successful getaways and busts and completion of career-specific milestones.
- We adapt the Hot Pursuit model to crowd working platforms : *Workers and requesters are the racers and cops*. Both classes progress along their careers in their own way. Workers gain **experience/rating** for **accurate and fast task completion**, while requesters earn it for **prompt payments, clarity of tasks and professionalism** in their work.
- Levelling up earns **more privileges** on the platform and unlocks **complex and higher-paying tasks**.

Level and Rating System

Experience points are earned through worker or requester specific actions. Earning a set amount of experience allows a worker or requester to **level up**. Levels help unlock *higher privileges and productivity tools*. These are indicative of the **quantity of work done/given.**

Rating is assigned to each worker or requester and is a function of several factors. Rating is indicative of **quality of work** done/given and **credibility**.

- Workers earn rating for accurate and fast completion of tasks, skill level and knowledge, time spent on platform, compatibility with requesters.
- Requesters ratings are similar to that on Turkopticon – generosity, promptness, professionalism, fairness and communicability, task clarity and feedback history.

Workers and requesters are classified into tiers based on experience and rating.

Worker tiers

Categorise workers into three tiers:

- **Novice or entry-level (Tier 3)** : Workers with a few days experience with the platform and just starting out. Lot of tasks might not be available to them at this point, since they are relatively unknown. Computer generated tasks whose solutions are known are provided to them – we can gauge their skillset, accuracy and efficiency. This acts as a initial seed.
- **Mature or intermediate (Tier 2)** : Workers who have established themselves on the platform and now qualify for taking on actual tasks. A machine learning algorithm interacts with their task selection choices and learns about their preferences. Preferences can also be manually tweaked. This helps generate HITs suited to them later on.
- **Professional or expert (Tier 1)** : Workers who have lot of experience with the platform and are focused on efficient completion of tasks. Some tools designed to increase productivity become available to them – next task becomes available immediately after completion of tasks without task selection screen. Machine learning data is used for task recommendation. Option to skip over to another task and disable this feature can be provided.

Requester tiers (Fill in if you have ideas)

Categorise requesters into three tiers:

- **Novice or entry-level (Tier 3)** : These are requesters who are just starting out with the platform and have no idea how much to pay for their tasks. Preferably, early requesters must work as a worker to understand the hardships faced and give due respect to the worker.
- **Mature or intermediate (Tier 2)** : Requesters who have understood the platform and are now capable of generating well-designed tasks with appropriate pay.
- **Professional or expert (Tier 1)** : Requesters who require large amounts of human computation and use the platform on a regular basis. Tools are made available to increase their productivity and help with the analysis of results.

Classification of Tasks

- All tasks are not the same, some require more effort than the other. Classification is essential if we attempt to **standardize price of tasks**.
- We planned to classify tasks first broadly like information retrieval, social media, surveys, OCR tasks. These could be further classified into simple and complex tasks and so on, to form a **ontology tree**.
- The ontology tree can be arranged/organised based on **nature of task** (broad categories listed above), **skillset** (mathematical/visual/logical) and **complexity**.
- The ontology tree classification of tasks draws inspiration from **Wikipedia**. **Moderators**/top-level workers/requesters are allowed to **add/reposition** task categories in the ontology tree.

Compatibility and Worker-Requester Social Graph

We use the Facebook social graph model to conceptualize worker-requester relations and compatibility. We have a crowdworker graph (bipartite for now i.e no relations within workers or requesters) whose nodes are requester and workers, and compatibility is modelled by the edges.

Compatibility plays a role in how visible a given requester's HIT is, to various workers. Workers having higher compatibility with the requester are more likely to see the HIT and therefore take it up.

Compatibility develops through a two-way feedback system between workers and requesters after completion of a task. It is centralised around the simple Yes/No question :

- Would you like to do a task from this requester again? [For workers]
- Would you delegate a task to this worker again? [For requesters]

Feedback form

- Feedback forms must be designed to be simple and as non-disruptive to the workflow as possible.
- Ideally only mouse interaction should be required to fill in the forms.
- Could be compulsory for Tier-2 and Tier-3 users and optional for Tier-1 users. Unfilled answers should not affect the rating of the requester/worker in any way.
- There will be a two-stage feedback – one during completion of task and the other, after its completion.

GUI Components

Simple push-buttons provide an objective answer to task parameter (say clarity). Can be good/neutral/bad.

Radio buttons to answer a simple Yes/No question.

Slider to rate a value.

Feedback form for workers

- First stage of feedback for workers involves them rating the clarity of the instructions, if the task was appropriately priced and whether optimum time was given for the task.
- Second stage involves requesters being rated on the promptness of their payment and their professionalism in general (mass rejections, feedback given to workers)

An upvote/downvote option inspired from Quora, signifies the overall experience of the worker with that requesters, querying whether he would like to work for said requester again. This can be implemented through a radio/flat button.

GUI Components

Simple push-buttons provide an objective answer to task parameter (say clarity). Can be good/neutral/bad.

Radio buttons to answer a simple Yes/No question.

Slider to rate a value.

Feedback form for requesters

- Requester feedback has only one stage which involves him rating his satisfaction with the submitted work. Promptness can be judged by the system. There is no during-task feedback.

A similar upvote/downvote option signifies the overall experience of the requester with the worker, querying whether he would like the worker to do another task for him.

GUI Components

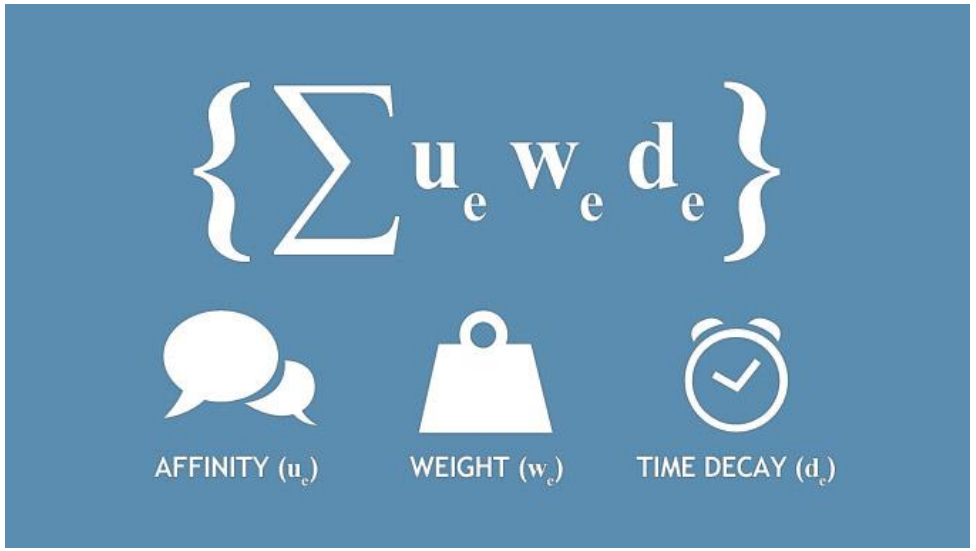
Simple push-buttons provide an objective answer to task parameter (say clarity). Can be good/neutral/bad.

Radio buttons to answer a simple Yes/No question.

Slider to rate a value.

HIT Visibility

HITs are analogous to posts on Facebook, liking or commenting a post is analogous to accepting a HIT by the worker.



On Facebook, the posts visible to users are determined by the EdgeRank algorithm, which is a function of affinity, weight and time decay.

On the platform we develop, the HIT visibility is determined by :

- **Worker-requester compatibility** determined from the Social Graph (Affinity)
- **Importance of task** to worker determined from skillset, task preference and time/money payoff (Weight)
- Tasks whose **deadlines** are approaching given more preference (Time Decay)

Finally a randomized algorithm scrambles and sorts these HITs and chooses which ones are visible to the worker.

Reducing Worker Idle Time

- A common issue on current crowdsourcing platforms is, certain weeks generate good payoff for workers while other weeks don’t. While there is no effective way to predict trends in task demand, we can look for compensation techniques.
- Auto-generated tasks whose solutions are known can be made available at all times. During idle scenarios (low task demand), these can be taken up for work.
- This enables workers to improve their experience and ratings during downtime, which in turn makes them eligible for tougher and more intricate tasks later on.

Worker and Requester Profiles

Requester and workers could have a profile page for purposes of communication and knowledge. This is similar to the concept of a wall on Facebook.

Requester profiles may carry information about rating, tier, level and experience points, typical allotted work, contact information (email/github/IRC) and perhaps a button to IM. An option to subscribe to this requester may be allowed – increase visibility of tasks from this requester.

Worker profiles may carry information about rating, tier, level, skillset and task preferences. Personal information is subject to disclosure. However personal information has to be provided in order to take up demographic-related tasks.

A short summary of this page can popup whenever the mouse is scrolled over a worker/requester name (similar to Turkopticon's feature)

Task designing and generation

Good task designs are characterised by :

- Goal of the task is understood quickly, maybe through keywords or short intro at the top.
- Task is well structured and formatted.
- Good entry-level barrier questions to reduce spam and malicious entries.
- Compactness/low redundancy of tasks – more information transferred through lesser microtasks
- Easy to evaluate (through tools) and easy to complete

In order to write good HITs quickly and easily, templates/wizards may be made available for use. There will be a set of official templates which are added by expert users, subject to formal testing; and a set of templates added by other users. This official/unofficial template model is similar to the homebrew/third party system of apps on a mobile phone.

Templates/wizard creators will be an embedded web app available to all users. Expert users have the privilege of pushing their creation to official status. Each template/wizard will have a rating associated with it, similar to app ratings – an expert and user rating; as well as a comments section.

Changing the Legal System

A common issue is work theft, where requesters do not acknowledge work submitted, however incorporate it in the results. This happens because requesters enjoy intellectual property rights over the work submitted and not the workers.

A legal solution might be tough to implement considering the vast multitude of countries workers come from.

A meet-in-the-middle workaround might be to allow workers to submit “bad feedback forms” to the requester with appropriate evidence attached. These might be analysed by moderators who may decrease the rating of the requester if found guilty. Workers may be penalised if found to submit false claims. This facility may be included as part of the terms and conditions of the platform.

DARK HORSE IDEA 1

Standardizing payment for similar task classes. Involves three issues – task classification, price allocation and price normalization.

Task classification database can be edited only by expert requesters/workers who can add new classes or reposition them.

Price allocation can be done using a free-market system. Suggest adding a question in feedback “How much would you have liked to work for?”, “How much would you be willing to pay” for workers and requesters respectively. Can be implemented using a slider in feedback. We construct several demand-supply curves for different performance tiers.
(To fit economic assumption of all goods equal/equal efficiency)

DARK HORSE IDEA 1

The different demand supply intersections can be grouped together to give an appropriate price range, with a price floor and ceiling. Floors and ceilings are required to allow flexibility in pricing for requesters (urgent tasks might invite desperate rates) and prevent exploitation/misuse.

Having a new currency system (Turkercoin) which is normalized for every country, constructed on the basis of minimum wage rates and poverty line statistics.

Also defocuses requesters from thinking in terms of their own currency and judge pricing in terms of a common rate.

DARK HORSE IDEA 2

Pre-employment tests

Business psychologists

These psychometric tests can be used to evaluate requesters and their nature, before they are allowed to join the platform. Can even act as their initial rating seed.

The other option is to let requesters work a specific time period as a worker, so that they appreciate the market and hardship faced by a worker, and delegate tasks accordingly.