Crowd Design

André van der Hoek
andre@ics.uci.edu

Crowdsourcing

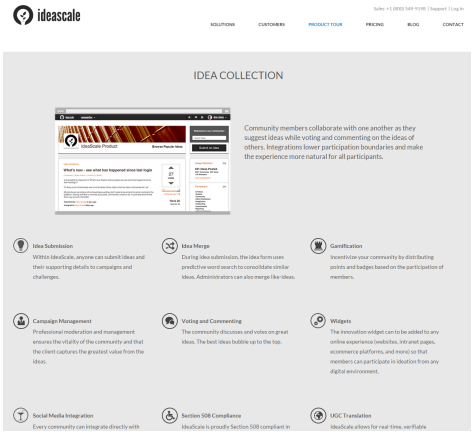
- “Crowdsourcing” is the act of taking a task traditionally performed by a designated agent (such as an employee or a contractor) and outsourcing it by making an open call to an undefined but large group of people.

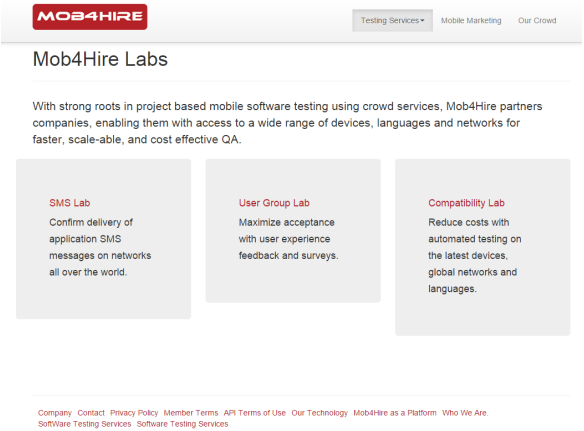
[Howe 2008]

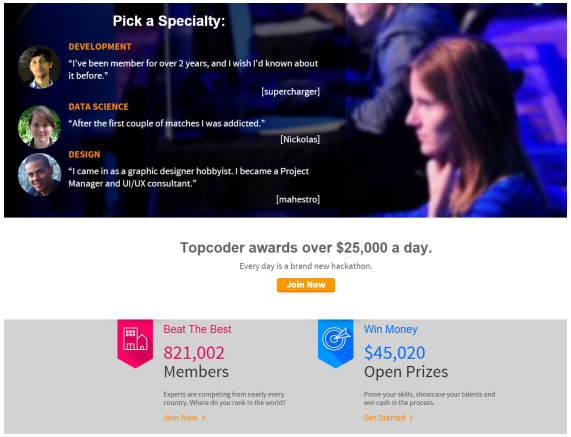
Potential benefits

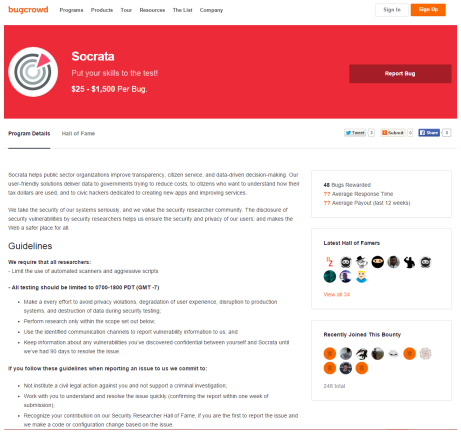
- Wisdom of the crowd
- Reduced time to market
- Generating alternative solutions
- Democratization of participation
- Employing freelance specialists
- Learning through work

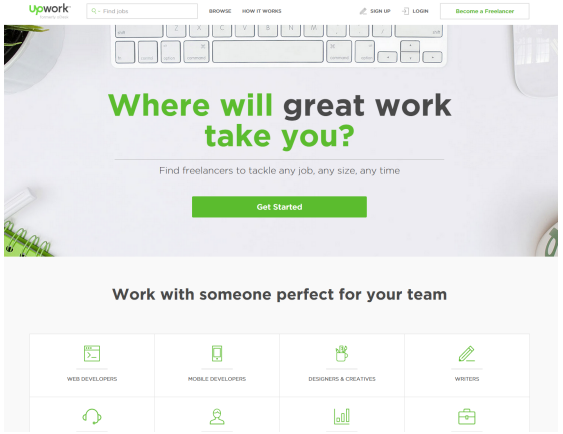
Example platforms

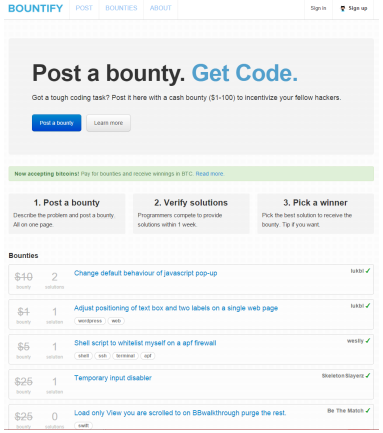












What we know today

- Various software engineering tasks *can* and *are* crowdsourced through a range of platforms
- Higher quality and less expensive code *[Lakhani, Garvin, Lonstein 2010]*
- Crowds are much smaller than anticipated; using crowds contributes to quality and creativity *[Wu, Tasi & Li 2013]*
- Waterfall model; best for less complex and stand-alone tasks; development costs much greater than expected; overhead much greater; quality issues pushed later in the life cycle *[Stol & Fitzgerald 2014]*

Two challenges

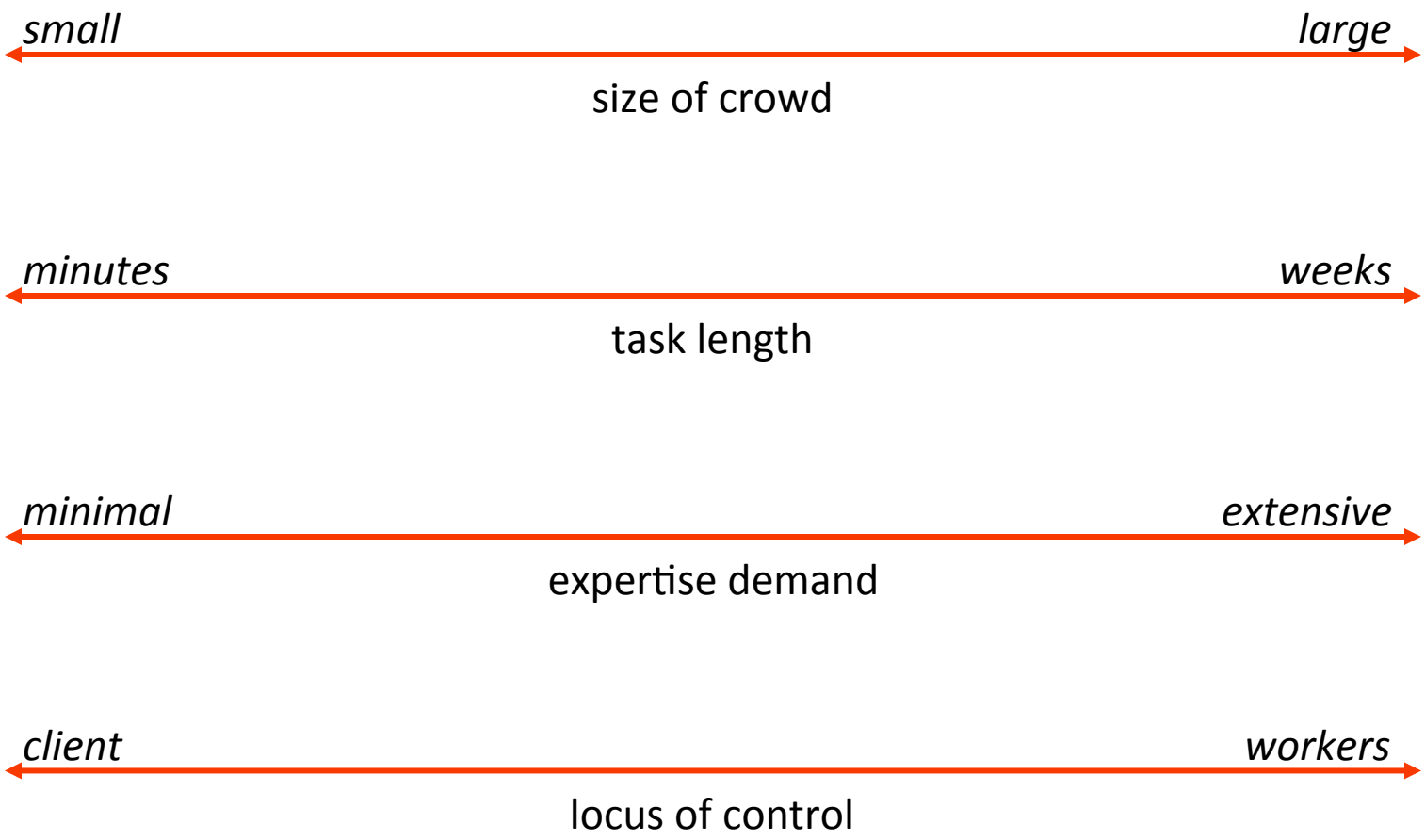
- Is it possible to crowdsource the more complex aspects of software development?
- Is it possible to crowdsource software development at massive scale?

Two challenges

- Is it possible to crowdsource the more complex aspects of software development?
- Is it possible to crowdsource software development at massive scale?

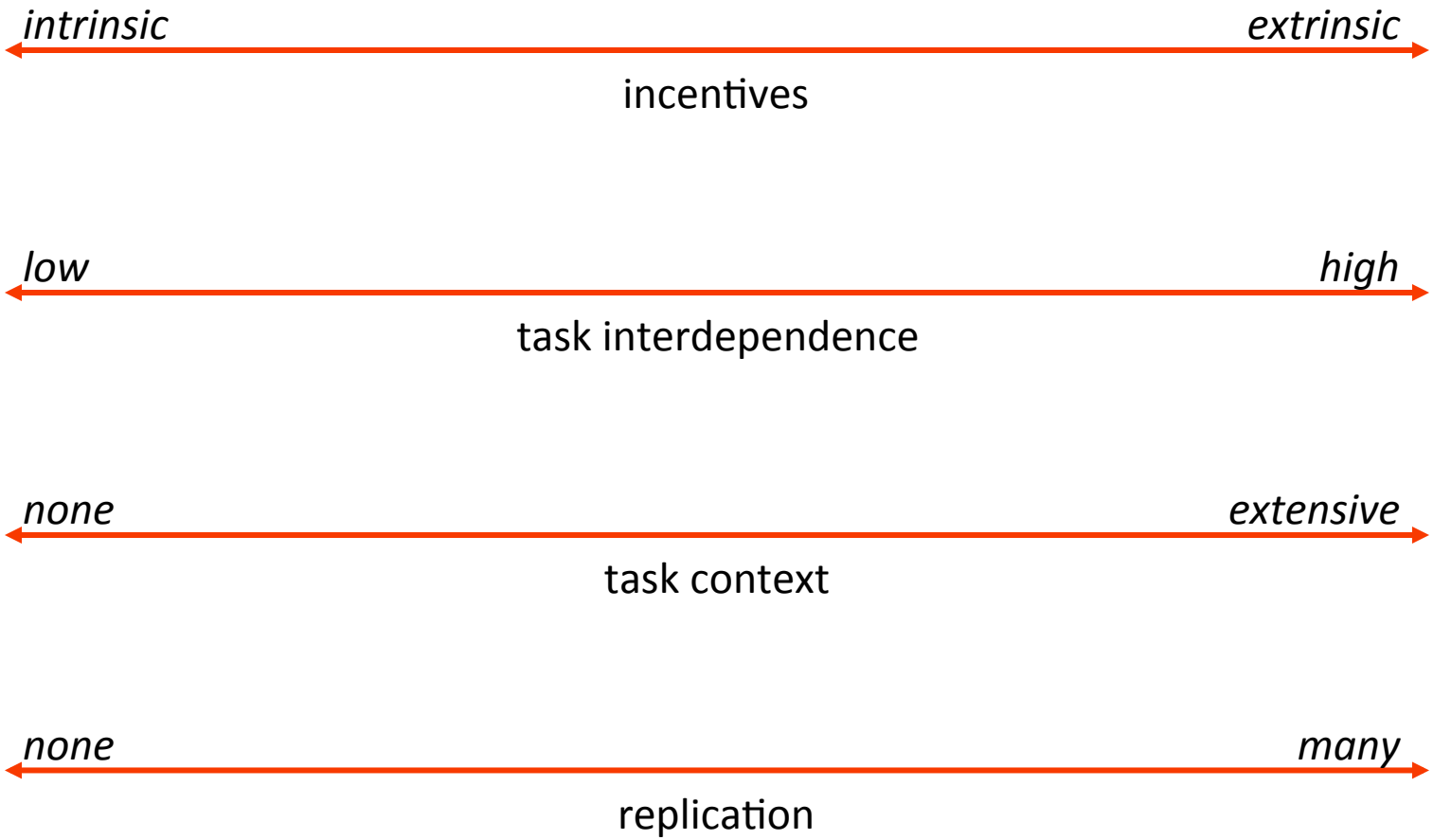
debugging programming design

“Essential” workflow design considerations



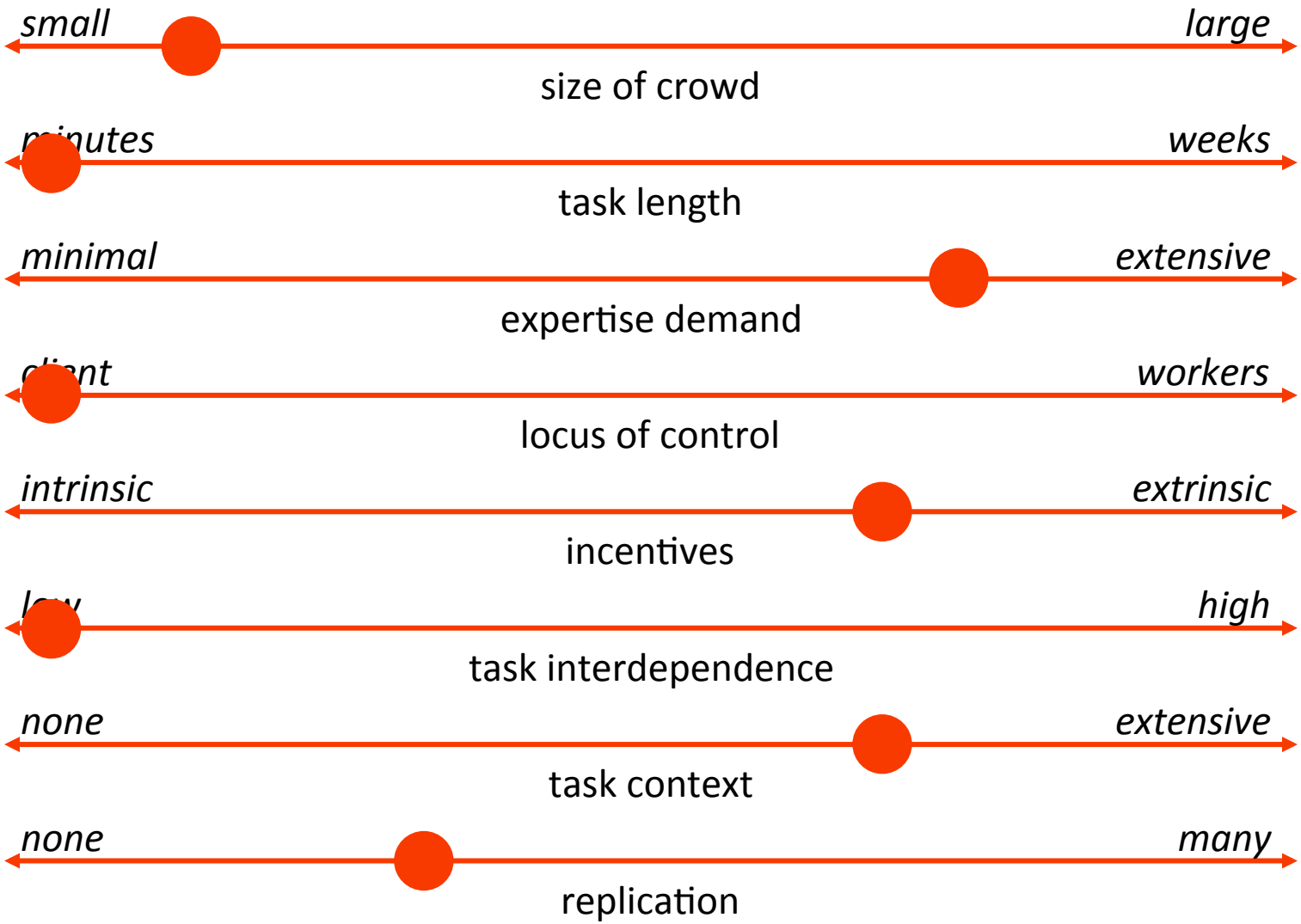
[LaToza & van der Hoek 2016]

“Accidental” workflow design considerations

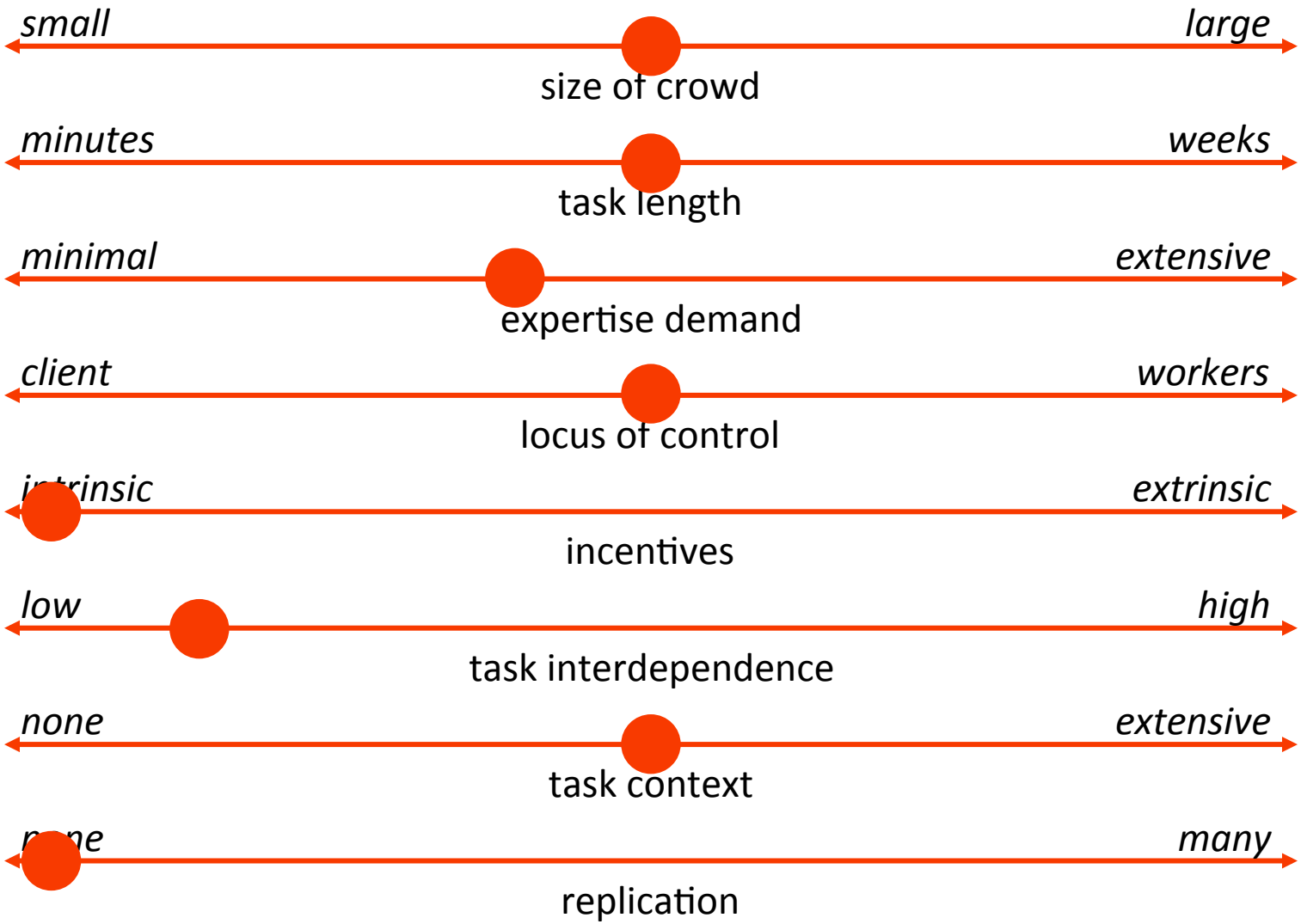


[LaToza & van der Hoek 2016]

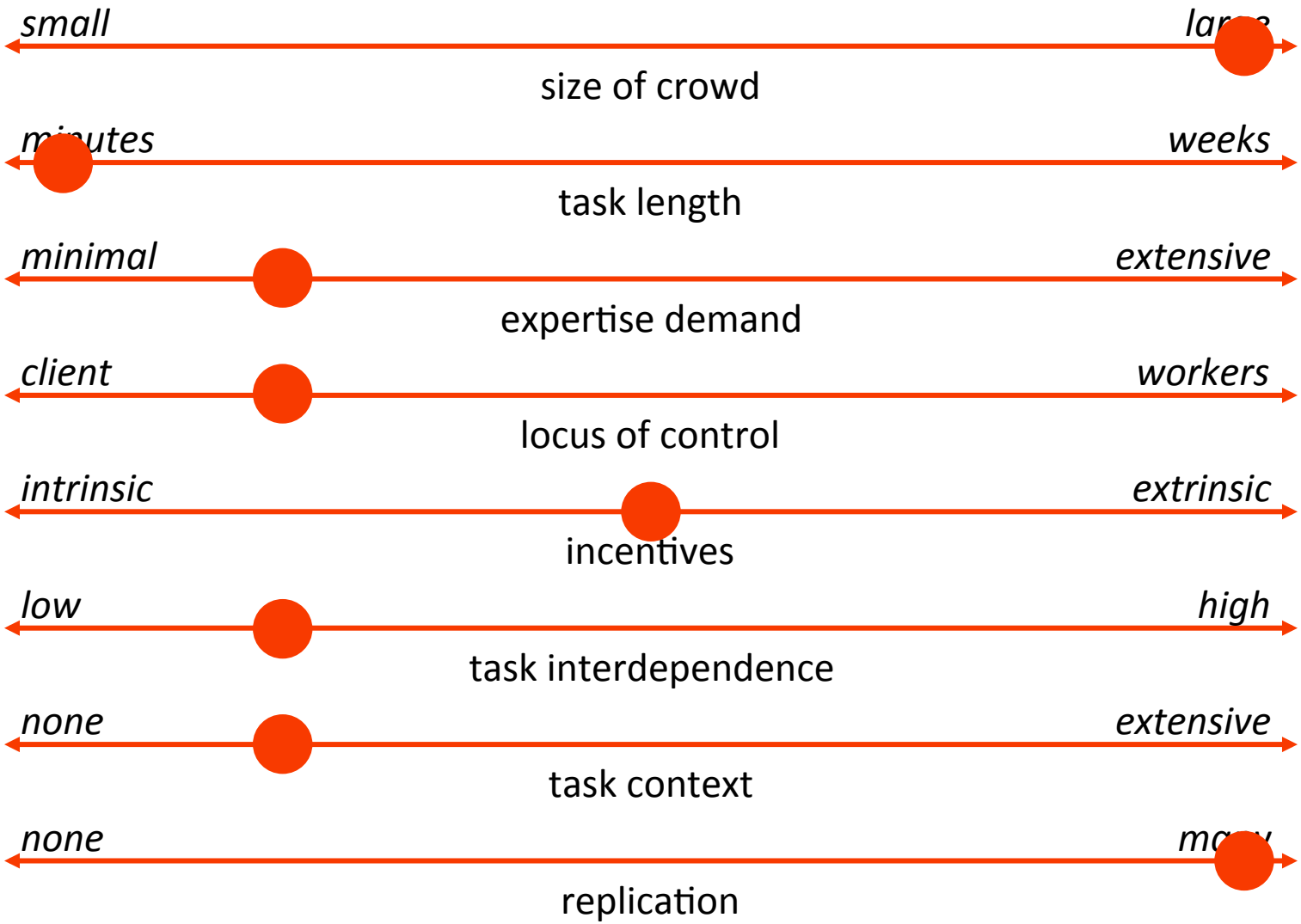
Competition



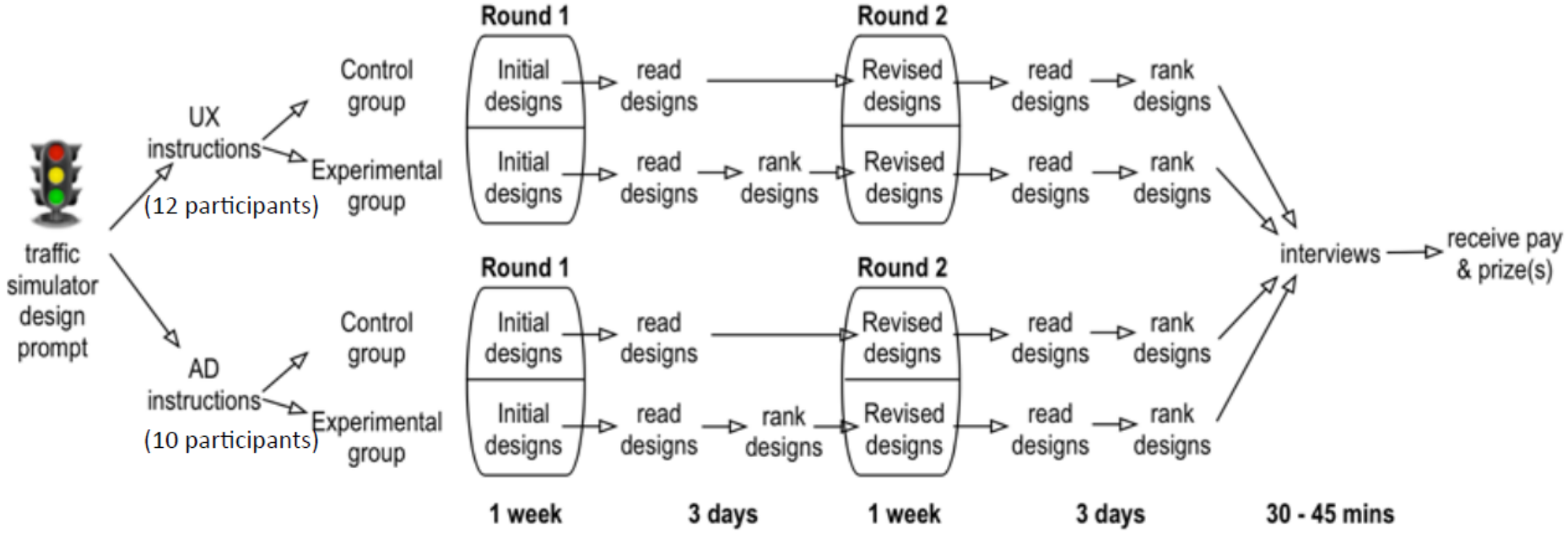
Collaborative community



Microtasking



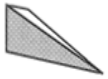
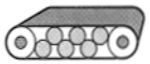
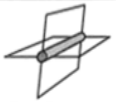
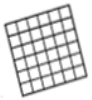
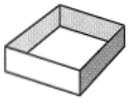
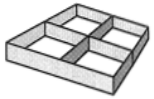
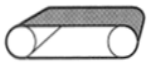
A first trial: two-phase competition



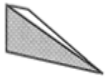
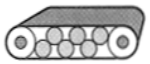
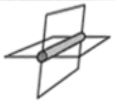
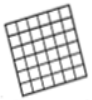
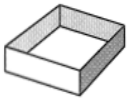
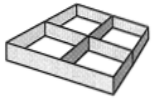
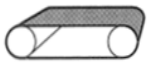
[LaToza, Chen, Jiang, Zhao & van der Hoek – ICSE 2015]

Second attempt: microtasking

Morphological chart

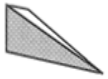
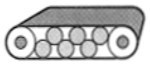
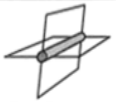
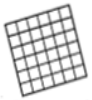
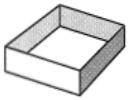
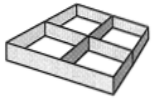
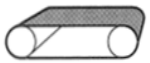
	Option 1	Option 2	Option 3	Option 4
Vegetable picking device		 Triangular plow	 Tubular grabber	 Mechanical picker
Vegetable placing device	 Conveyor belt	 Rake	 Rotating mover	 Force from vegetable accumulation
Dirt sifting device	 Square mesh	 Water from well	 Slits in plow or carrier	
Packaging device				
Method of transportation		 Track system	 Sled	
Power source	Hand pushed	Horse drawn	Wind blown	Pedal driven

Microtasking a morphological chart

	Option 1	Option 2	Option 3	Option 4
Vegetable picking device		 Triangular plow	 Tubular grabber	 Mechanical picker
Vegetable placing device	 Conveyor belt	 Rake	 Rotating mover	 Force from vegetable accumulation
Dirt sifting device	 Square mesh	 Water from well	 Slits in plow or carrier	
Packaging device				
Method of transportation		 Track system	 Sled	
Power source	Hand pushed	Horse drawn	Wind blown	Pedal driven

Can a crowd identify key decision points?
Can a crowd identify solution alternatives?
Can a crowd identify a complete design?

Microtasking a morphological chart

	Option 1	Option 2	Option 3	Option 4
Vegetable picking device		 Triangular plow	 Tubular grabber	 Mechanical picker
Vegetable placing device	 Conveyor belt	 Rake	 Rotating mover	 Force from vegetable accumulation
Dirt sifting device	 Square mesh	 Water from well	 Slits in plow or carrier	
Packaging device				
Method of transportation		 Track system	 Sled	
Power source	Hand pushed	Horse drawn	Wind blown	Pedal driven

Can a crowd identify key decision points?

Can a crowd identify solution alternatives?

Can a crowd identify a complete design?

Tool

Task:

Design an interface mechanism through which users build maps with roads and intersections.

Sketch solutions that cover the following requirements:

- The user can create a simple visual map of roads on an empty, rectangular canvas.
- The user can create a map that supports at least 6 intersections.
- Roads may only lead to 4-way intersections (3-way intersections are not allowed).
- The user can create a map that allows roads of varying lengths, with different arrangements of intersections.

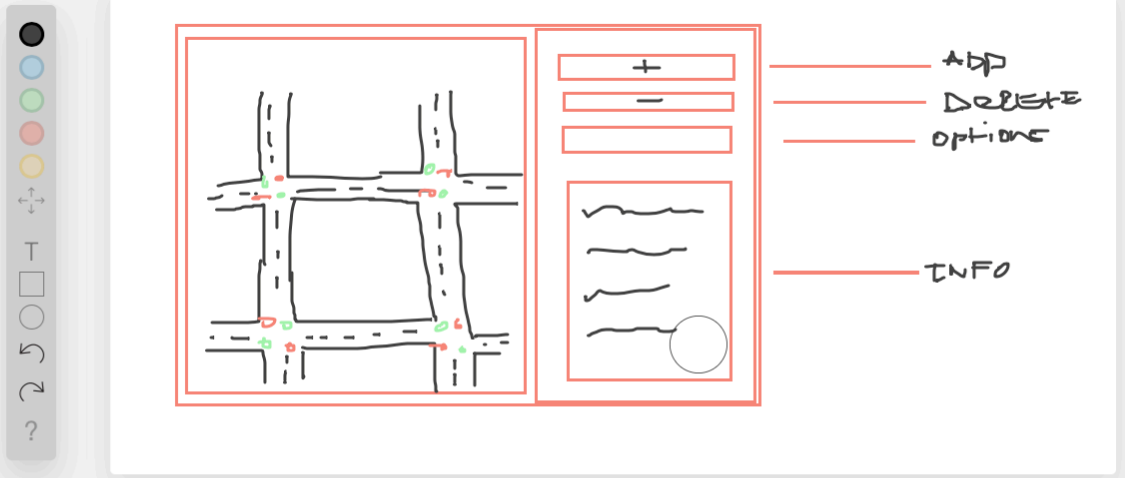
Tips:

- You don't need to support very complex maps. Try to focus on the different user interactions your solutions need to have to satisfy the requirements.

Reminder:

We are not looking for one perfect design but are interested in a variety of designs that each can have their own pro's and con's.

REVIEW & SUBMIT
TIPS
TASK INFO
QUIT



Name your first solution

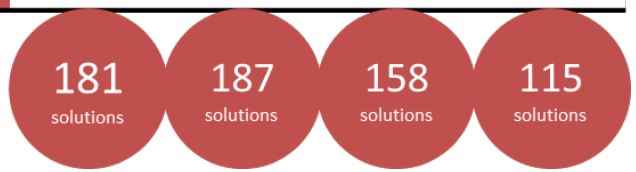
Explain your first solution

Two experimental conditions

- User interface design versus internal code design
- Workers work in isolation versus workers see the completed work of others

Participation

	User Interface (UI)	UI with examples	Internal Design (ID)	ID with examples
Signed consent form	1069	875	1474	836
Quit before taking qualifica-tion test	205 (19%)	191 (22%)	375 (25%)	229 (27%)
Failed the test	580 (54%)	431 (49%)	886 (60%)	456 (55%)
Passed the test	284 (27%)	253 (29%)	232 (16%)	151 (18%)
Entered the platform	282 (26%)	225 (26%)	220 (15%)	151 (18%)
Submitted completion code	88 (8%)	87 (10%)	94 (6%)	72 (9%)
Accepted work	78 (7%)	80 (9%)	76 (5%)	66 (8%)



Example solution alternative

name: Graphical representation of traffic at 6 traffic lights

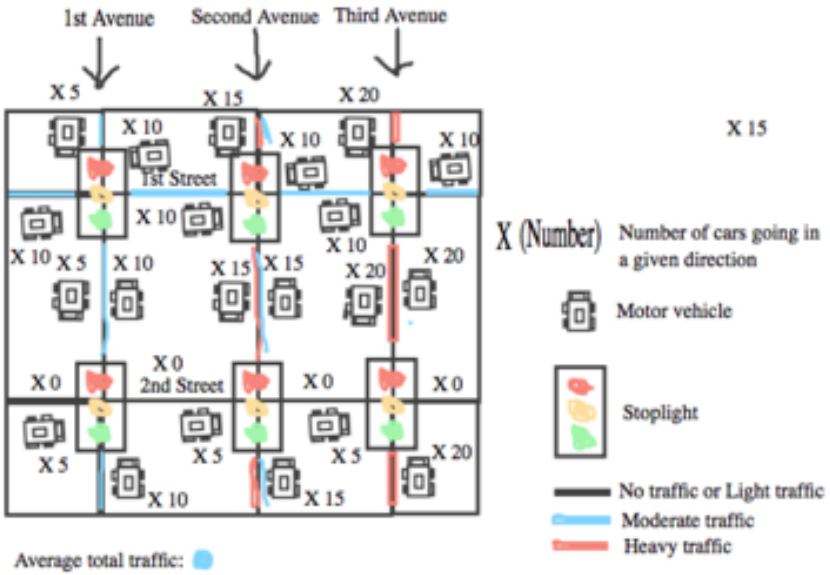
explanation: I decided that the user has to see what is going on in order to make a decision about what to do with the traffic light timings. So, I decided to graphically represent it in a way that would make it easier to see what was happening.

time spent: 42m

time spent name: 0s

time spent explain: 0s

workerId: SNktPPM3GkPX573K3



Example solution alternative

name: Graphical Traffic Flow Editor

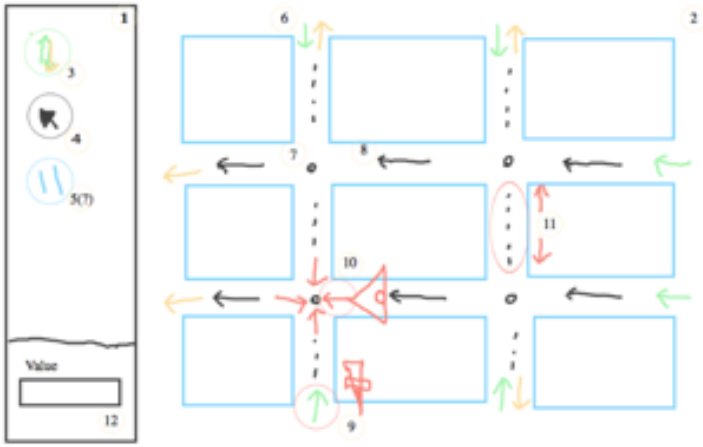
explanation: I goofed and drew a 3x3 grid with 4 intersections instead of a 3x4 grid with 6, but the principles should hold up the same regardless of map size or shape. (1) - Toolbar (2) - Map editing area (3) - Tool for placing entrances and exits, possibly left-click to place entrance, right-click to place exits or something to that effect (4) - Typical object select tool (5) - Tool for placing roads. I wasn't sure whether this interface is intended to be able to construct maps or simply change the traffic flows on existing maps, so functionality of this tool won't be expanded on much. Maybe a simple line tool type mechanism, where you click to place two nodes or select existing intersection nodes and a road is created between them (6) - Green arrows represent placed entrances and yellow arrows represent exits (7) - Intersection nodes. Should be automatically defined wherever two roads intersect. Maybe there could be a tool to manually place them as well? Not sure what impact or use that could have, except the possibility to create "garages" in areas between intersections, which I'll go over later (8) - Marker representing a segment of road. Dashed lines in this sketch represent 2-way streets, lines with arrows are one-way (9) - When an entrance is selected, a slider appears allowing for adjustment of the amount of traffic that will enter through that entrance. Maybe it also makes the arrow grow bigger&#x2Fthicker the higher the factor so that it's easy to tell at a glance the relative traffic of all entrances without having to select each one and check (10) - When an intersection is selected, arrows appear from each oncoming traffic direction (that one on the left side isn't actually valid because it's a one-way street. It probably wouldn't be a bad idea for the system to detect this, but it shouldn't do any harm if it doesn't since that arrow will simply affect the Zero cars coming that way). When you select an arrow, a UI element appears that allows you to adjust the probability that approaching cars will head in a certain direction. In this case if the slider is moved to one of the points of the triangle, that means a 100% chance cars will move in that direction, while leaving it in the middle of the right face of the triangle means an equal probability of any direction. This could be expanded for an N-way intersection to be any (N-1)-gon, but that could get messy and unintuitive pretty quick. Then again, anything more than a 5-way intersection is pretty unlikely in a realistic road map. Turn probability should NOT overwrite one-way streets however, unless a goal of the program is to model reckless/law-breaking drivers as well. (11) - When a road segment is selected, you can click and drag parallel to the road to make it a one-way street. A one-way street can be made two-way again by dragging in the opposite directon. Alternatively there could actually be two UI arrows with a dot in between. Possibility for more traffic flow control here, but most of what I can think of is already covered by entrances and intersections. (Other than vehicle speed, maybe? That's something I didn't consider at all until just now, hmm...) (12) - As with most editing programs, the option to look at and manually edit the actual values of whatever you're adjusting would be provided too, so you can be more precise with your flow settings if you like. -- Not shown in the sketch, but there could be a possibility of adding entrances and exits in between intersections as well, essentially garages on the blocks in between. These would simply be placed with the entrance/exit tool by clicking on the desired spot. Entrances would have the same entrance rate control slider, but in addition both entrances and exits could have an intersection-like probability controller that determines which way cars will turn when coming out of entrances, and the likelihood that passing cars will turn to take an exit they pass by. -- The ability to place entrances and exits and one-way streets so freely certainly presents the possibility of the user inducing conflict, but that might not be such a bad thing - If the map breaks somewhere, you should wind up seeing where it breaks sooner or later. But there could also be some kind of conflict handler, that's not really the part I'm meant to be worrying about here anyway. -- This interface might not be the most precise but it should be fairly approachable and intuitive to use. Theoretically there shouldn't be any major issues with non square-grid-based road systems either, and curved roads should be handled in the same way as straight road segmentsas far as directions, intersections, etc

time spent: 17m

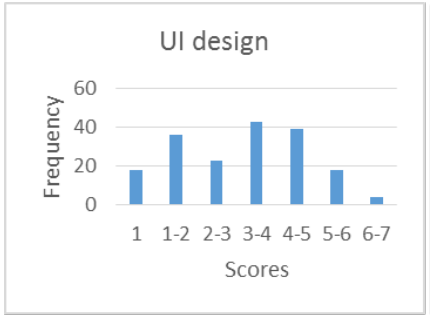
time spent name: 6s

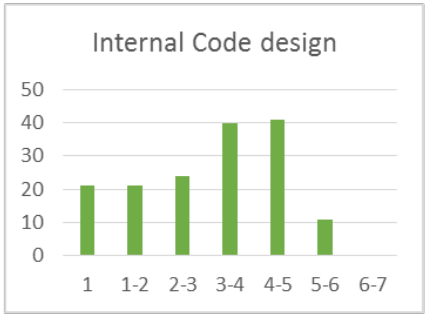
time spent explain: 0s

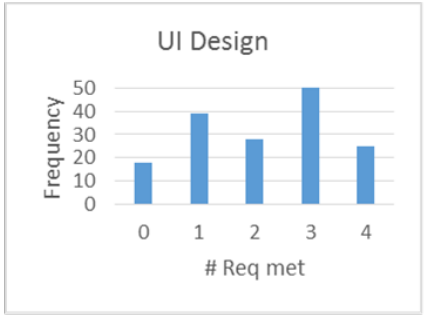
workerId: GiEkyu3uK4eeE9ujP

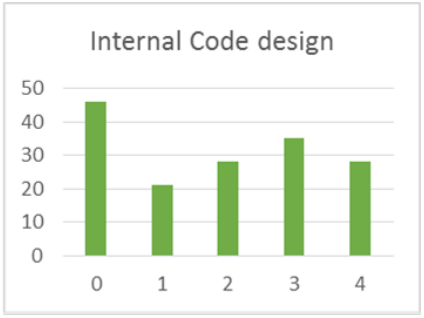


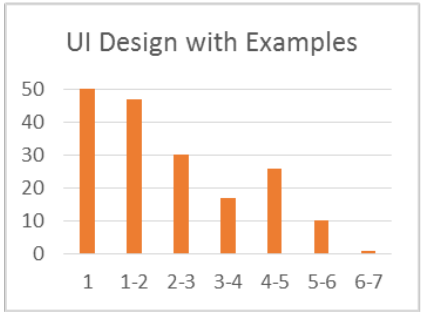
Quality

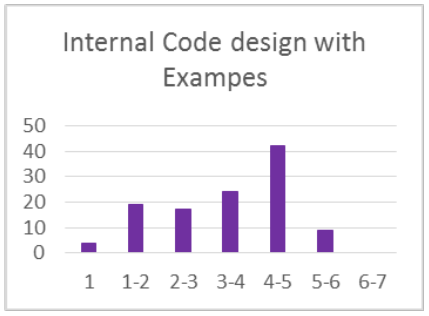

UI design
Frequency
60
40
20
0
1
1-2
2-3
3-4
4-5
5-6
6-7
Scores

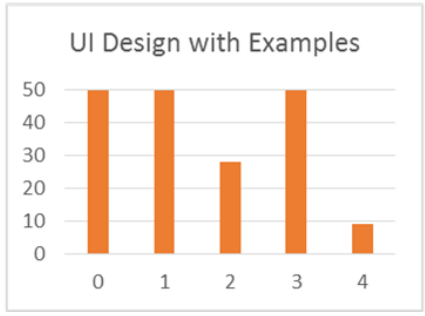

Internal Code design
50
40
30
20
10
0
1
1-2
2-3
3-4
4-5
5-6
6-7

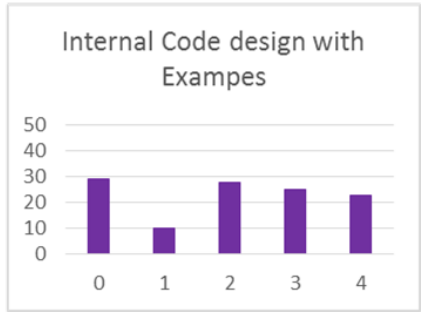

UI Design
Frequency
50
40
30
20
10
0
0
1
2
3
4
Req met


Internal Code design
50
40
30
20
10
0
0
1
2
3
4


UI Design with Examples
50
40
30
20
10
0
1
1-2
2-3
3-4
4-5
5-6
6-7


Internal Code design with Exampes
50
40
30
20
10
0
1
1-2
2-3
3-4
4-5
5-6
6-7


UI Design with Examples
50
40
30
20
10
0
0
1
2
3
4


Internal Code design with Exampes
50
40
30
20
10
0
0
1
2
3
4

Diversity

	User Interface Design		
	No Examples #categories	With Examples #categories	Total
task 1.1	11	11	13
task 1.2	13	10	13
task 1.3	10	7	10
task 1.4	14	9	15

	Internal Code Design		
	No Examples #categories	With Examples #categories	Total
task 2.1	10	11	12
task 2.2	10	7	10
task 2.3	10	8	11
task 2.4	14	13	15

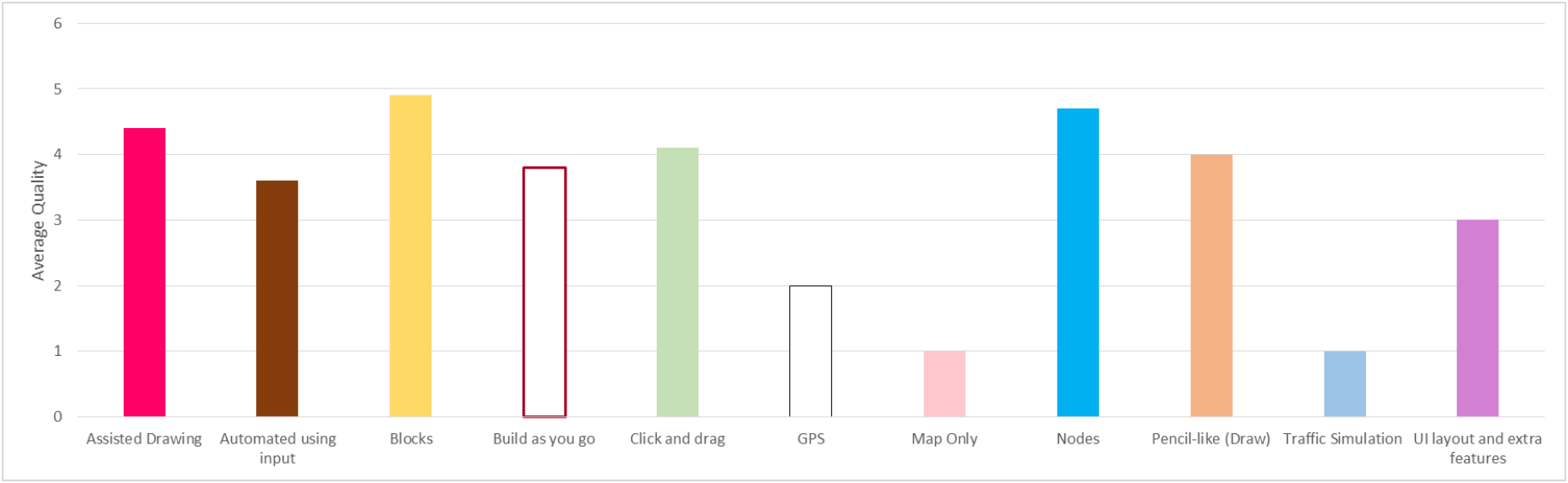
Overall diversity versus individual diversity

	decision point description	#categories
dp 1	map creation	11
dp 2	setting of traffic light timings	13
dp 3	determining the flow of traffic	10
dp 4	visualization of the state of the simulation	14

Worker ID	Category				
Worker MB1	Blocks				
Worker MB2	Click and drag	Build as you go	Blocks	GPS	
Worker MB3	Traffic Simulation				
Worker MB4	Blocks	Assisted Drawing			
Worker MB5	Click and drag	Pencil-like (Draw)	Pencil-like (Draw)	Pencil-like (Draw)	UI layout and extra features
Worker MB6	UI layout and extra features	UI layout and extra features	Automated using input	UI layout and extra features	UI layout and extra features
Worker MB7	Map Only	Map Only	Map Only	Map Only	Map Only
Worker MB8	Build as you go				
Worker MB9	Nodes	Nodes			
Worker MB10	Click and drag	Click and drag			
Worker MB11	Pencil-like (Draw)	Blocks	Blocks	Build as you go	
Worker MB12	Click and drag	Click and drag			
Worker MB13	Map Only	Map Only	Map Only	Map Only	Map Only
Worker MB14	Nodes	Nodes	Click and drag		
Worker MB15	UI layout and extra features	UI layout and extra features	UI layout and extra features	UI layout and extra features	Click and drag
Worker MB16	Click and drag				
Worker MB17	Assisted Drawing	Assisted Drawing	Pencil-like (Draw)		
Worker MB18	Blocks	UI layout and extra features			
Worker MB19	Blocks	UI layout and extra features	UI layout and extra features	UI layout and extra features	UI layout and extra features
Worker MB20	Map Only				
Worker MB21	Build as you go	Blocks	Pencil-like (Draw)		

Quality across categories

Average quality score per category (Map creation)



Conclusions

- It is feasible for a crowd to generate a diverse range of solution alternatives
- Solution alternatives, however, vary strongly in quality, with only a moderate number that are of sufficient quality
- It is important for diversity to involve multiple workers; individual workers do not create diverse sets of solution alternatives
- Displaying examples has a negative effect: diversity goes down slightly and quality goes down significantly

Future work

- Repeat the experiment, only displaying examples of sufficient quality (done)
- Repeat the experiment, with a different crowd (Topcoder)
- Address research question #1: generating decision points with a crowd
- Explore hybrid models