



Panacea's Cloud

Contextual Geotracking of
Incident Markers for Disaster
Search-and-Rescue Operations

Ev Cheng and Kourtney Meiss

The Problem

When disasters strike and there are mass casualties it is difficult to properly triage patients and keep track of them. This becomes exponentially difficult when there's a lack of nearby infrastructure.

Panacea's Cloud: What is it?

A system designed for managing mass casualty incidents

used during mass



CONTAMINATED

Personal Property Receipt: [Barcode]

Evidence Tag: [Barcode]

Destination: [Barcode]

File: [Barcode]

TRIAGE TAG [Barcode]

S L U D G E M

AUTO INJECTOR TYPE: [1-3]

AUTO INJECTOR TYPE: [1-3]

Respirators: [1-3]

Decontamination: [1-3]

Mental Status: [1-3]

Other: [1-3]

VITAL SIGNS

Temp	BP	Pulse	Respiration

CONTAMINATED

EVIDENCE

MORGUE

IMMEDIATE Life Threatening Injury	IMMEDIATE Life Threatening Injury
DELAYED Non-Life Threatening Injury	DELAYED Non-Life Threatening Injury
MINOR Walking Wounded	MINOR Walking Wounded

EVIDENCE

CONTAMINATED

MORGUE

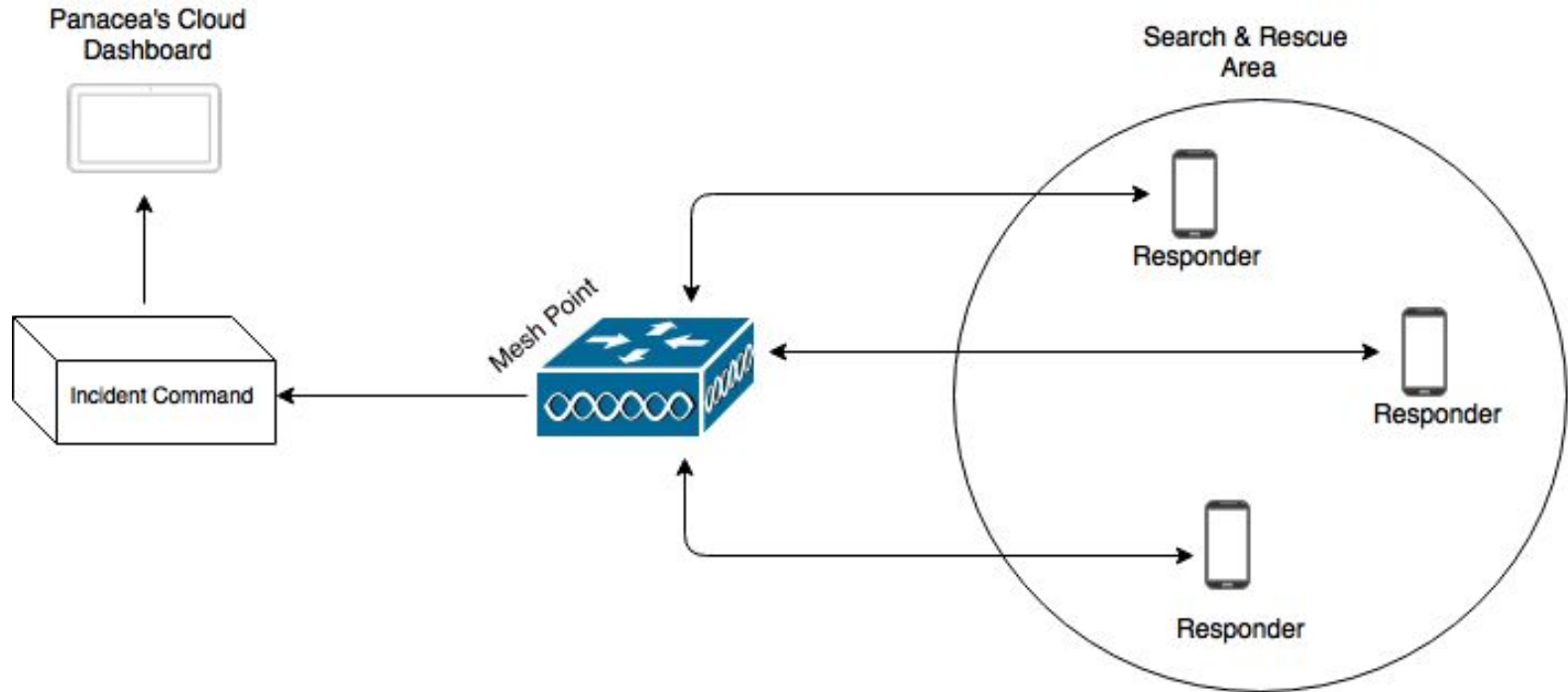
Pulseless/Non-Breathing

IMMEDIATE Life Threatening Injury	IMMEDIATE Life Threatening Injury
DELAYED Non-Life Threatening Injury	DELAYED Non-Life Threatening Injury
MINOR Walking Wounded	MINOR Walking Wounded

CONTAMINATED



Panacea's Cloud: Components



Panacea's Cloud: Significance

Increased communication in disaster scenarios can help incident commanders gain situational awareness to best allocate resources.

Decreased triage time means more lives can be saved.

Existing

- Uses one type of communication
- Uses
- Relies



Figure 3. Responder Mapview

DIODAMA

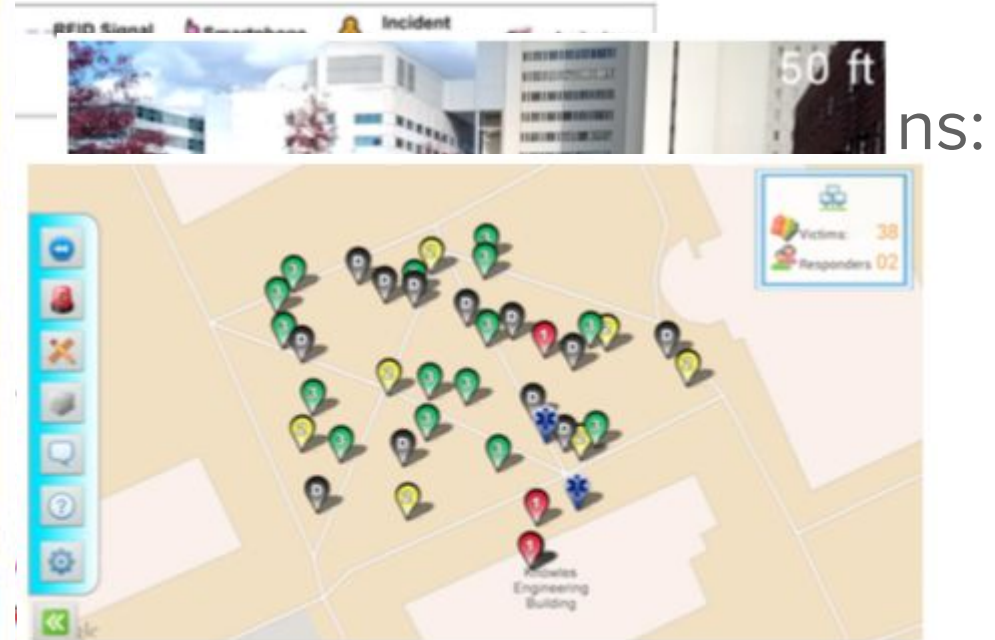


Figure 2. Disaster Zone Mapview

Technical Tools

- Javascript, CSS, HTML
- React and JSX
- Redux

Research Question

- How do we represent space and time on a static map and make it usable?
- During a crisis people must be able to take in past and current events in a single glance
- Must be presented with low cognitive burden

OKR (1)

Design a data model for data monitoring by building a filter framework around the data sources (e.g., low resources). Allow for the future addition of data sources and different actions.

OKR (1)

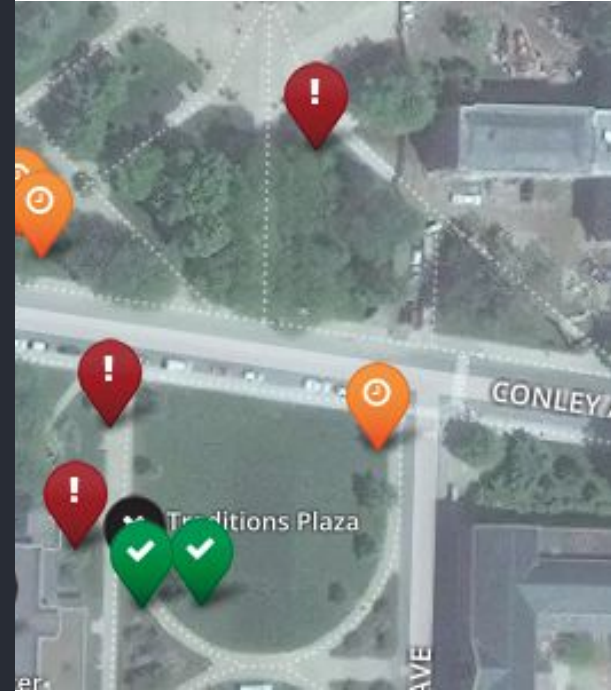
KR: Mapping resource supply in relation to time and space

~~KR: Integrate multimedia uploads with map~~

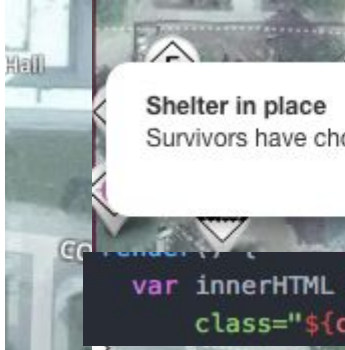
Mapping Space



```
{  
  "markers": {  
    "patient": {  
      "deceased": {  
        "color": "black",  
        "faIcon": "fa-close",  
        "iconColor": "white",  
        "shape": "circle"  
      },  
      "immediate": {  
        "color": "red",  
        "faIcon": "fa-exclamation",  
        "iconColor": "white",  
        "shape": "circle"  
      },  
      "delayed": {  
        "color": "orange",  
        "faIcon": "fa-clock-o",  
        "iconColor": "white",  
        "shape": "circle"  
      }  
    }  
  }  
}
```



Custom



```

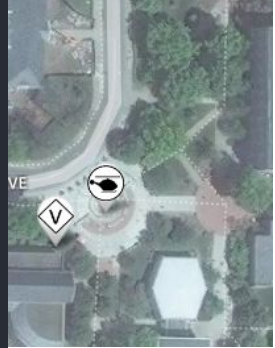
"1": {
  "img": "01.png",
  "name": "Structure No Damage",
  "description": "Low Risk, low probability of further collapse"
},
"2": {
  "img": "02.png",
  "name": "Structure Damaged",
  "description": "Medium Risk, structure damaged"
},
"3": {
  "img": "03.png",
  "name": "Structure Failed",
  "description": "High Risk, may be subject to sudden collapse"
},
"4": {
  "img": "04.png",
  "name": "Structure Destroyed",
  "description": "Complete destruction of structure"
}

```

```

var innerHTML = `})

```



Symbol	Name		
	Structure No Damage	Lo	
	Structure Damaged	M	
	Structure Failed	Hi	
	Structure Destroyed	Co	
	Assisted	M	
	Evacuated	Su	
	Rescued	Technical rescue that required physical intervention	Custom 6

Mapping Space in Temporal Frame

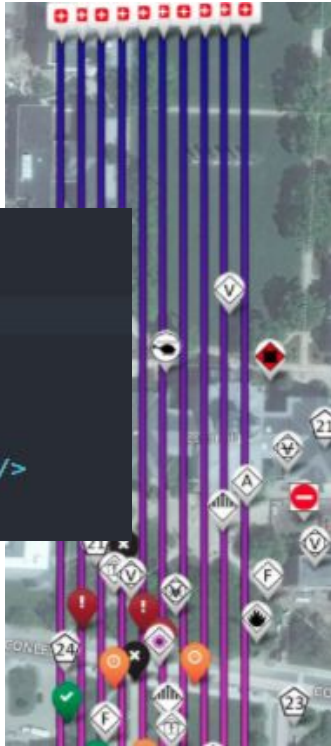
```
responders: [ {  
  id: <integer>,  
  firstName: <string>,  
  lastName: <string>,  
  latestGeotag: <Object>,  
  geotags: {  
    <geotagId>: {  
      id: <integer>,  
      responderID: <integer>,  
      lat: <float>,  
      lng: <float>,  
      timestamp: <DateTime>,  
      type: "responder"  
    }  
  }  
}
```



Mapping Space to Relative Time



```
palette={ { 0.0: '#e835ca',  
            0.15: '#ad0051',  
            0.33: '#ff738f',  
            0.48: '#cccc18',  
            0.66: '#008f15',  
            0.79: '#35e8ca',  
            1.0: '#1400ad' } } />
```



Spatiotemporal Filter



DISPLAY TIME

07/14/2016 01:14

FILTER START TIME

07/09/2016 20:49

FILTER END TIME

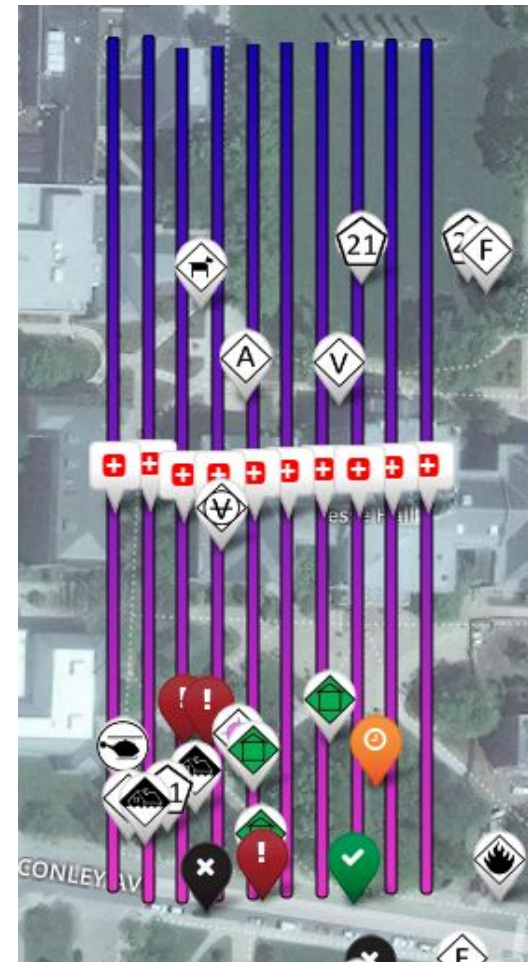
07/18/2016 21:06

ABSOLUTE START TIME

07/01/2016 12:00

ABSOLUTE END TIME

08/12/2016 03:00



Time Slider

```
,  
[CHANGE_DISPLAY_TIME]: (state, action) => {  
  return Object.assign({}, state, {  
    displayTime: action.payload  
  })  
},  
[CHANGE_START_TIME]: (state, action) => {  
  return Object.assign({}, state, {  
    startTime: action.payload  
  })  
},  
[CHANGE_END_TIME]: (state, action) => {  
  return Object.assign({}, state, {  
    endTime: action.payload  
  })  
},  
},
```

```
const mapActionCreators = {  
  togglePanelCollapse: () => togglePanelCollapse(),  
  changeDisplayTime: (val) => changeDisplayTime(val),  
  changeStartTime: (val) => changeStartTime(val),  
  changeEndTime: (val) => changeEndTime(val),  
  changePlay: () => changePlay(),  
  getNewIncident: () => getNewIncident()  
}  
  
eFi  
tog }  
s.props.toggleStart(new Date(parseInt(e.target.value[0], 10)))  
s.props.toggleEnd(new Date(parseInt(e.target.value[1], 10)))  
if playing, pause.  
this.props.playBoolean){  
his.props.togglePlay()  
  
changes displayTime based on filter boundaries  
this.props.startTime >= this.props.displayTime){  
his.props.toggleDisplay(new Date(this.props.startTime))  
  
else if(this.props.endTime <= this.props.displayTime){  
this.props.toggleDisplay(new Date(this.props.endTime))  
}  
}
```


Change of Redux Store

```
[{"first_name": "Kendall", "last_name": "Park", "id": 2, "trackee_type": "responder"},  
{"first_name": "Josiah", "last_name": "Burchard", "id": 3, "trackee_type": "responder"},  
{"first_name": "Lukie", "last_name": "Pookie", "id": 4, "trackee_type": "responder"},  
{"first_name": "Kourtney", "last_name": "Wortney", "id": 5, "trackee_type": "responder"}]
```

```
[{"id": 210, "timestamp": "2016-07-08T14:32:32.757Z", "lat": 38.9183502218025, "long": -92.361735674789, "trackee_id": 35, "tracker_id": null, "tracker_mac": null},  
{"id": 209, "timestamp": "2016-07-08T14:32:32.742Z", "lat": 38.962046324691, "long": -92.3164487515913, "trackee_id": 34, "tracker_id": null, "tracker_mac": null},  
{"id": 208, "timestamp": "2016-07-08T14:32:32.727Z", "lat": 38.9123869855167, "long": -92.3273838133057, "trackee_id": 33, "tracker_id": null, "tracker_mac": null},
```

```
return apiCall(options)  
}
```

```
responders: [ {  
  id: <integer>  
  firstName: <string>  
  lastName: <string>  
  geotags: [ {  
    id: <integer>  
    responderID: <integer>  
    lat: <float>  
    lng: <float>  
    timestamp: <Date>  
    type: "responder"  
  }  
]  
}]
```

New

Previous

Refactoring Code

```
findResponderGeotags(responder, geotags){  
  return _.filter(geotags, function(tag){  
    return responder.id == tag.trackee_id  
  })  
}
```

```
const mapStateToProps = (state) => ({  
  panelCollapsed: state.map.panelCollapsed,  
  ...state.map.geotags, ...state.map.trackers
```

```
// based on startFilter and endFilter, returns an array of [lat,long,z] for Track  
getTrackCoordinates(start, end){  
  var tags = this.props.geotags  
  var coords = []  
  for(var i=0; i < tags.length; i++){  
    if(Date.parse(tags[i].timestamp) >= Date.parse(start)  
      && Date.parse(tags[i].timestamp) <= Date.parse(end)){  
      coords.push([tags[i].lat, tags[i].long, Date.parse(tags[i].timestamp)])  
    }  
  }  
  return coords  
}
```

Reset Button

Do you want to reset?



Markers

```
export function fetchNewIncident() {
  return {
    [CALL_API]: {
      types: [INCIDENT_NEW_REQUEST, INCIDENT_NEW_SUCCESS, INCIDENT_NEW_FAILURE]
      endpoint: '/incidents/n reset() {
      schema: Schemas.INCIDENT if(!this.props.isResetting){
      method: "GET"           this.props.toggleReset()
      }
      }
    }
  }
}
```

RESET

Task

- Fire
- sp
- 28
- We
- tec

US&R Response System Task Forces



age
county
es




ion



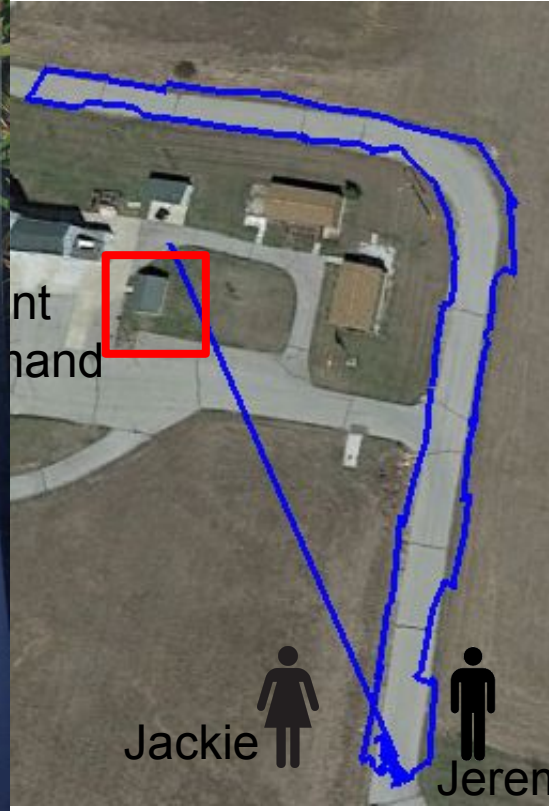
Task

FEMA News Photo <https://www.fema.gov/photosdata/low/52864.jpg>



- Structural Assessment Icon (Structure Failed)
- Victim Detected by Canine & Confirmed by Camera (Victim Confirmed)
- Physical Rescue Completed (Rescued)

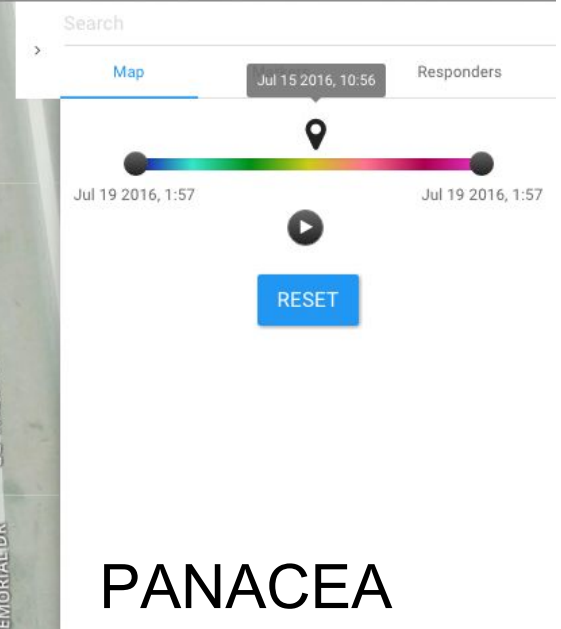
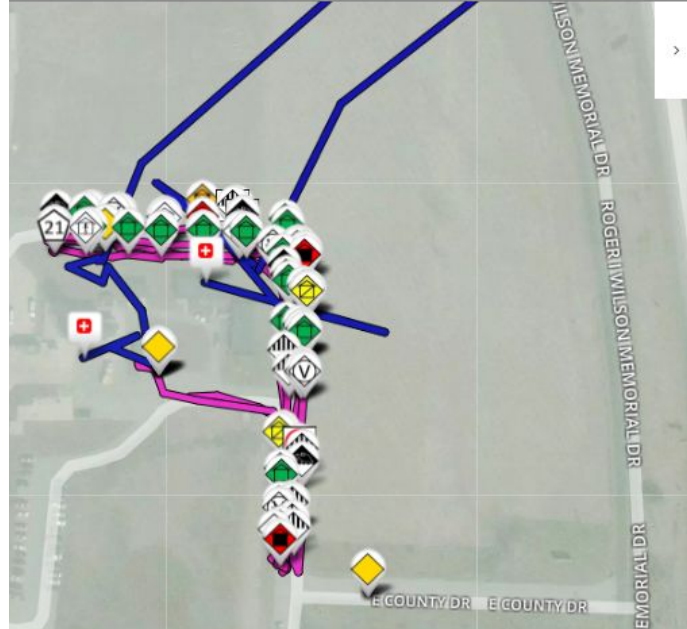
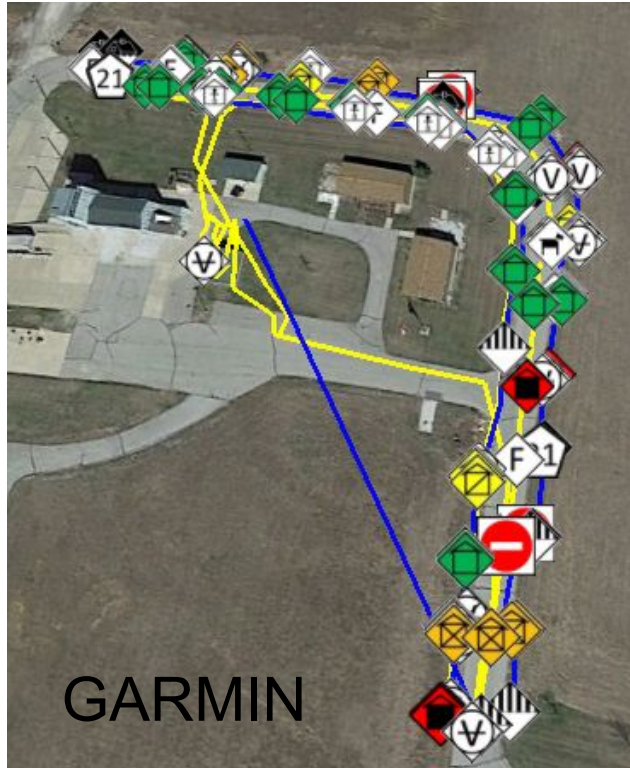
*Complete search marking sticker per ROE



Task Force 1- Recon Jets

- Relied on the internal camera to send QR codes about which marker was selected
 - Photo exposure didn't work well -- all white in the sunlight or all black in the shade
- Doesn't have the best GPS
- Not a competitive market right now
- Didn't throw the idea away, but phones make more sense to use right now

Task Force 1- Results

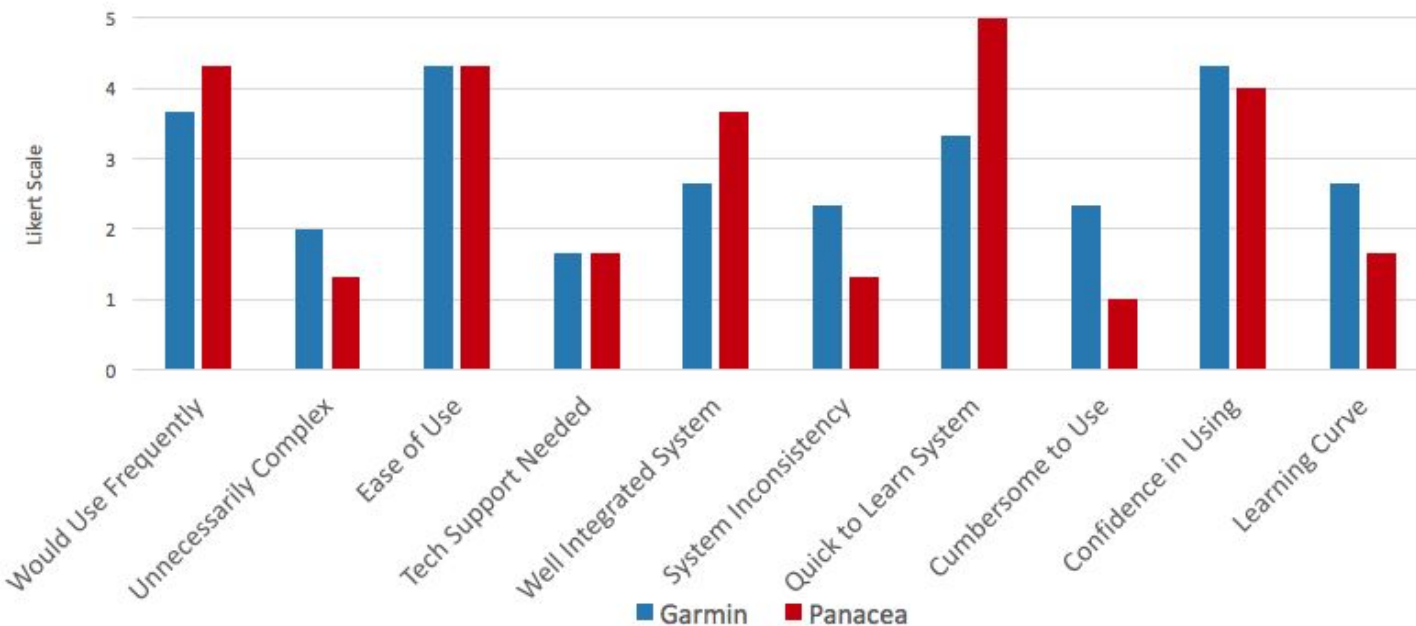


Task Force 1- Usability Questions

- I think that I would like to use this system frequently
- I found this system unnecessarily complex
- I thought this system was easy to use
- I think that I would need the support of a technical person to use this system
- I found the various functions in this system well integrated
- I thought there was too much inconsistency in the system
- I would imagine that most people would learn to use this system quickly
- I found this system very cumbersome to use
- I felt very confident using this system
- I need to learn a lot of things before I could get going on this system

Task Force 1- Usability Questionnaire Results

Panacea vs. Garmin Usability



Future Work

- How do we filter data and present it with a recommendation for a non-manually triggered action, so that the user can best proceed?

Future Work

Recommend intelligent actions.

KR: Design recommendations for intelligent actions based on research

KR: Design a framework for a notification system

Future Work

- Symbol multiplier for high entry numbers
- Tally of each kind of marker
- Add confirmation that marker has sent
- Display date and missions under incident tab

Conclusion

- Questions?