RESPONSIBLE HUMANITARIAN & DISASTER RESPONSE PROJECT LIFECYCLE

GWOB.org
GeeksWithoutBounds

COMPILED:
Willow Brugh | GWOB | @willowbl○○

WRITTEN:
Lindsay Oliver | GWOB | @RosalindOfArden

INTERVIEWED:
Lisha Sterling | GWOB | @lishevita
Sara-Jayne Terp | Open Crisis + Ushahidi | @bodaceacat
Heather Leson | Humanitarian OpenStreetMap Team | @HeatherLeson
Max Shron | www.shron.net | @mshron

DESIGNED:
Margaret Killjoy | @magpiekilljoy

CREATE WITH FORETHOUGHT

FINITE PROJECT LIFE SPAN

So, you've got a thing you want to make. Before you start, before one line of code, graphic, or prototype comes into being, adhere to one of the first rules of project building; **have a kill date**. This keeps you on schedule, mindful of time spent, and focused on the metrics of success for your project within its finite life span.

Tools that achieve this:

- Google Calendar's "how many times to repeat" option on recurring events.
- Mailing lists remind you about your subscription.
- For institutional review boards (IRBs) in academia, data plans are mandatory, and include the following considerations:
    - What data do you collect?
    - How do you store it?
    - How long do you store it?
    - How do you destroy it?
    - What rights do your test subjects have for access?

WHY ARE YOU BUILDING THIS?

In order to build a tool that is useful and responsible, certain requirements for your project must be met in order to merit its creation.

- What is the justification for the tool you are building?
- What need is being met?
- Are there other tools out there that do the same thing?
  ...if so, is it addressing the potential user's needs?
  ...if not, how will your tool achieve that?
- How will you contribute to existing code bases?
- How will your tool get into the hands of the users?
- Do your intended users have input into the design and a way to give feedback (co-design model)?

WHAT COUNTS AS "DONE"?

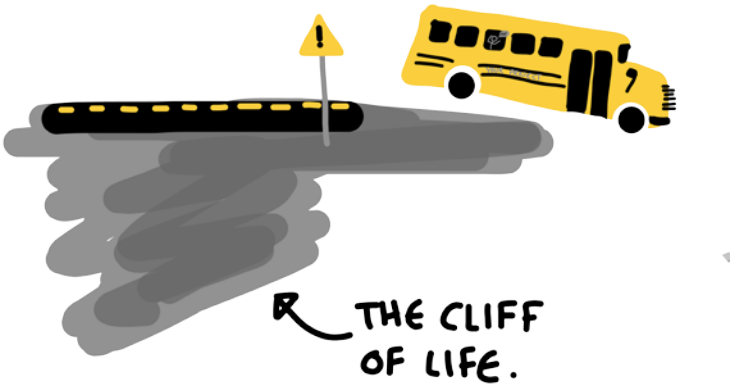
If your tool serves a specific purpose, is there a point where it's fulfilled its intended purpose? Form a plan for that moment, and communicate with users regarding current usefulness, re-purposing of the tool for related uses, and its effectivess. Ask for user feedback on the tool during the crisis or time of specific intended use.

PLAN FOR FAILURE

Any process involves something going wrong. Planning for that should be a part of any project. Plan for how the project will dissolve in the event that it fails (ownership of resources, data responsibility, business closure). Do this **before** failure occurs, and it will mitigate hurt feelings. Communicate this plan to your users so they aren't blindsided by a sudden loss of a tool, and are able to warn the project team of potential issues in advance. This plan should be housed with all of your documentation.

Tools for planning:

- DIYTOOLKIT.ORG
- WIKI.USHAHIDI.COM/DISPLAY/WIKI/USHAHIDI+TOOLKITS
- WIKI.USHAHIDI.COM/DISPLAY/WIKI/10+QUESTIONS+YOU+SHOULD+ASK
- WWW.SLIDESHARE.NET/USHAHIDI/USHAHIDI-TOOLBOX-REALTIME-EVALUATION
- WWW.SLIDESHARE.NET/USHAHIDI/USHAHIDI-TOOLBOX-ASSESSMENT
- BLOG.USHAHIDI.COM/2013/12/11/USHAHIDI-3-0-ALPHA-RELEASE



THE BUS PROBLEM

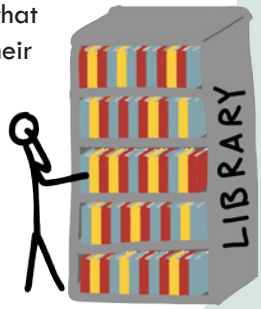
When the sequester forced cutbacks in NASA's program in Spring 2014, the yearly International Space Apps Challenge was affected by the temporary reduction in staff. The event website had bugs, glitches, and dead links among other issues. No staff/knowledge base redundancies were in place to continue project efforts in the event that personnel were reduced. Therein lies the "Bus Problem." If you were hit by a bus, would your project stand on its own? Would it be able to continue if key members of your team were to leave/be incapacitated? Do you have a kill switch or a "keep alive" switch? These redundancies and core questions need to be addressed in order to appropriately plan for the efficacy and long-term viability of your project, and also to determine when it's time to pull the plug.

BUILD ON OPEN SOURCE

Don't build something from scratch. Myriad Free and Open Source Software (F/OSS) solutions to difficult coding challenges already exist and can be implemented by tweaking and optimizing for your specific project needs. Included in these solutions are encryption models/standards and rigorous data structures. Use them, and document your code for use in future F/OSS projects in the original central repository. Pushing your updates back to that library helps future projects succeed, and can engage users to contribute to bug fixes and add new features to your own repository. Resources such as The Humanitarian Toolbox or Github maintain a store of existing projects and code. Project teams can pull resources and repurpose as needed (with appropriate and lawful attribution). Check out Stop Hacking Without Specing [TEXTONTECHS.COM/2013/11/STOP-HACKING-WITHOUT-SPECING-A-TOP-10-NEEDED/] for more information.

OPEN DATA AND INFORMED CONSENT

You're making a tool with or for users; it follows that you should ask them what they'd like done with their data. Surveys, interviews, polling, and more can aid your team in refining exactly what users want and need. Don't assume you know what's best for a particular population. They know themselves and their needs far better than you do. Go in with the intent to learn, not teach. Note what



informed consent looks like in different cultures and places where your tool will be deployed, and adjust accordingly in compliance with local, regional, and national laws and customs. Putting your project and data collection/usage plan into a plain language disclosure [HACKATHONFAQ.COM/CONSENTFORM] will go a long way toward communicating your intent. It will also gain the trust of your users that you are acting in their best interest and with their explicit consent. Include a Threat Model [WWW.OWASP.ORG/INDEX.PHP/THREAT_RISK_MODELING] of your data before you push it live, and communicate those risks with your userbase well in advance of deployment for their review.

CONSIDERATIONS FOR OPEN DATA

If what we do is minimize risk by providing information, we also have to minimize risk in releasing the data.

– Sara-Jayne Terp

Oftentimes, data is thrown away because people don't know where to put it, or worse, personally-identifying information is released (including mapped data). In the first hour of a crisis, all minimization of data risk goes out the window due to the overwhelming need to provide safety for those at risk. In order to responsibly provide life-saving information during a crisis while protecting the identities of affected populations, critical analysis and care is a **must** in how the data is presented. For example, in the Ushahidi platform you can only download the reports on data, not the geographic information. Putting data out in aggregate reduces the risk of individuals being targeted and their data being used for nefarious purposes.

However, there are risks in providing any type of information. In the Aleppo governorate of Syria, publishing mapping data on bread lines in the rebel side of the city might result in bombing by the government. In NYC during Sandy Response, mapping shelters/crisis centers could inadvertently result in victims going to dangerous locations or not finding a shelter, as they often change locations during crises due to damage, need, etc.

MITIGATING DATA RISK

It is **always mandatory** to ask these questions about your potential data risk:

- Could this data point be exploited for evil, and how?
- Do the potential exploiters have the resources, desire, and knowledge to use it for evil?
- Can the good that releasing this data does outweigh that potential evil?

We've learned from railway engineering that *everything can be a risk*. There is no zero-risk event, only very low probability/low cost events (Note: Risk is defined as a combination of probability and severity of outcome – usually as a multiplication of the two – which is why every death has a $ cost in risk calculations that can vary based on context). Know ahead of time who will use the tool and data. Ask **more than one** person in that set of users how it could be used in other situations. Ask other people from other deployments. Ask for feedback from organizations like Crisis Mappers [CRISISMAPPERS.NET], Ushahidi [WWW.USHAHIDI.COM], and Humanitarian Open StreetMap [HOT.OPENSTREETMAP.ORG] Look for people with knowledge/experience of the same geographic and subject matter area, and preferably someone who's done a similar risk analysis before but has some distance from your current calculation. Plumb their networks and contacts to glean context, considerations, and cultural differences to be aware of pre-deployment of your tool.

If you're modifying existing data, contribute that data back to the original repository, community, and population the data is about (if possible) as you would with modified code. If you want to conceal certain sensitive information, it becomes increasingly difficult as the parameters of sensitive data increases. Standby Task Force, a cadre of digital responders [BLOG.STANDBYTASKFORCE.COM], has a plethora of datasets publicly available in Google spreadsheets with built-in attribution. You should create an up-front data agreement that all volunteers sign. It should be the same as a F/OSS agreement, and should state that you're contributing *for use* under some form of open data license. This type of license does not guarantee that the date will be released, but that it is possible to do so. Use the ODbL license as a template/boilerplate [OPENDATACOMMONS.ORG/LICENSES/ODBL], and additionally implement a "do no harm" principal into the agreement.

Additionally, there are tools to clean datasets available, as well as training programs to facilitate safe data cleansing practices. For example, The Eric & Wendy Schmidt Data Science for Social Good Sum-

mer Fellowship [DSSG.IO] created the Ushine tool for dataset cleansing [BLOG.USHAHIDI.COM/2013/08/20/USHINE-FROM-THE-DATA-SCIENCE-FELLOWS].

WHAT IF I'VE ALREADY BUILT SOMETHING?

If you've already begun building, there are tactics you can use to protect your users and keep documentation on a project in order to learn from it. Ushahidi is still learning lessons from their Legacy platform, and continues to analyze the structures, failures, and successes to keep future projects rigorous. However, you may not need to keep *all* the data from your archived projects. For example, why keep people's phone numbers, when all you need to know is that several messages came from the same person? Instead, keep a unique generated "ID" for that number (though consider keeping the area code). Replace full names with unique IDs, etc. Sometimes it's enough to keep high-level data such as graphs, summaries, statistics, and/or category lists.

HOW TO KNOW WHEN TO KILL YOUR PROJECT

Are people not using your project anymore? Has it gotten into the Wrong Hands? Kill it off. There is no reason to keep a project if it's unusable or potentially harmful. Have a clear explanation of why you shut the project down in advance of the actual shutdown in a place that isn't the back woods of your own wiki. You want to keep your users informed about the status of a tool they may depend on, even if it's only a few users.

"Kill off your project"

- You spend more time feeling guilty about not working on the project than you spend time working on the project.
- You have free time but you don't want to work on it at all.
- "The potential result no longer yields a net profit (inserting your respective economy). Put it on a burning ship to sea." – Ben Moore
- The money runs out.

Once the project is closed, the data can be published as a file on your site with a project overview. The overview can

serve as a resource for those attempting a similar project so that they don't have to do extensive research or start from scratch.

– Amber Case
caseorganic.com/blog/2010/07/track-your-happiness-results-emotional-feedback/

If you're keeping raw Twitter data, have an agreement on what you can't keep it for. If you've completed the project and want to archive the findings from it, make it useful for others to learn from; create a categories list, compile screenshots, and document any processes and useful tips you've learned along the way. Dump the personally-identifying information (PII), unlicensed data, and stale data such as field hospital lists so it isn't used for current work. **Caveat: This is for local deployments only. Remote data should be transferred to the actual deployment locations so they can make the decision on what's right for them and their data.**

I DIDN'T ASK FOR CONSENT FOR EXTENDED USE OF DATA

It's important to note what information is actually sensitive. There's a fuzzy line between what is personally identifiable and what is useful as open data. In smaller populations, even a very short list of indicators can make triangulating and identifying individuals incredibly easy. Evaluate whether linking the identity to the trait is important – and it usually isn't. *Consider that it takes a very small number of data points to identify almost anyone on the planet.*

Data is hard to repurpose, especially if you don't have the first license on it. *The currency in these projects is trust*, and your contributors trust you with their information. The moment you give that information to an outside source, trust in you is broken, along with the whole community.

Review the Crowd Globe project's parameters on data storage [irevolution.net/2013/02/03/crowdglobe]. Many in the response community consider this to be data hoarding with total disregard to the rights of the original populations. Those who create the maps are responsible for their usage, which means monitoring how they can be used to manipulate and exploit people. Discussions and analyses of tactics and ethical data storage/publishing guidelines can be found

via a discussion group [BIT.LY/USHWORLD] and the Ushahidi Crowd-Map site [WORLDUSHAHIDIS.CROWDMAP.COM]. Note that these forums are in-progress drafts of metadata standards and processes, and as such should not be used seen or used as currently deployed standards or finalized directives.

OPEN SOURCE YOUR PROJECT!

Read Geek's Without Bounds' guide to understanding Open Source and safety in disaster and humanitarian work [GWOB.ORG/BLOG/2013/12/09/IS-OPEN-SOURCE-SAFE-FOR-CIVIC-PROJECTS-AND-DISASTER-RESPONSE].

HAND-OFF

In order to do a successful hand-off of a project, create the following for your project successors' benefit:

- Incident plan / after action plan.
- One-pager on how to properly close the project.
- Templates of all tasks and a how-to on what those processes entail.

WHAT IF SOMEONE IS USING IT NEFARIOUSLY?

You can't stop someone from using your created technology for alternate purposes. You can stop your own development. The better thing to do in those cases is to keep working on it and, if possible, write patches that make it difficult to use the project in the nefarious way.

One way to reduce the possibility of nefarious use is to build 'good' into your code designs. For instance, a system that relies on a connected, cooperating community is difficult for a hierarchical, fixed-mind military to greatly exploit (e.g. less manpower, imagination, etc). You can't stop nefarious action, but you can certainly discourage and slow it down.

ON THE META LEVEL

Embed a "kill date" on your **platform**. If people are using that platform, this becomes a part of the community and culture. Alternatively, create a set of stages for the platform. For example, a crisis platform could have the following stages: initial situation awareness,

crisis response, early recovery, recovery, and handover. Each of these stages have different information needs and different/progressively more restrictive rules that can be applied. Stages can have expected transition dates relative to each and informed by the unique situation needs. The most important lesson to learn is that there is no easy mandate. Each event will change the needs/time required to complete tasks, and is informed by engaging and communicating with all portions of the community (mappers, in-field deployments, affected populations, etc.).