An 18F How-To on Evaluating Software Purchase Decisions

Whether to Build, Buy or Borrow?

Alicia Rouault
Steven Reilly

Traditionally, Three Options

1 Build

Custom Build, *in-house*

2 Build

Custom Build, *outsourced* labor

3 Buy

Commercial Off the Shelf (COTS) Buy and/or Buy Software as a Service (SaaS)

There is usually a fourth

4 Borrow!

For some software components or specific needs you may have, you can reuse, or redeploy existing free, open source code from another government or organization

PostgreSQL

Drupal™

ckan

Some examples of open-source projects commonly used in government

When should you decide whether to build or buy?

When you can describe what users need with enough detail to perform market research

This might happen quickly, or it might take some time

18F

What sort of details are needed?

User needs (at a high level)
Policy requirements
Budget range
External integration requirements
Procurement/security lead times

18F

Step 1: Fully understand what you need

Goal: Enough discovery to be able to articulate the goals of the system you think you need, and a general idea of some of the components

18F

Step 2: Prioritize your needs

Goal: Understand what matters to you most. Weigh multiple factors like budget, time, user needs and how easy would it be to buy?

18F

Evaluation Tool: What's Most Important?

Trade off Sliders

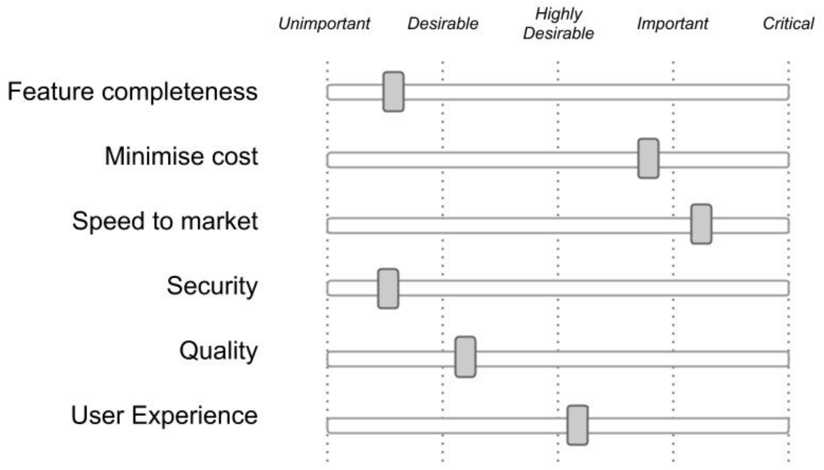
Discuss trade-offs as early as possible, and revisit them as often as necessary

Don't just stick to the generic list of sliders - **make them as specific to your project as you can**

Factor these into your estimates and your project approach

Communicate - make sure that everyone around the project sees the same picture

Always keep your trade-offs in mind when making decisions (big or small), and discuss them openly



http://www.blackpepper.co.uk

Evaluation Tool: Weighing multiple factors

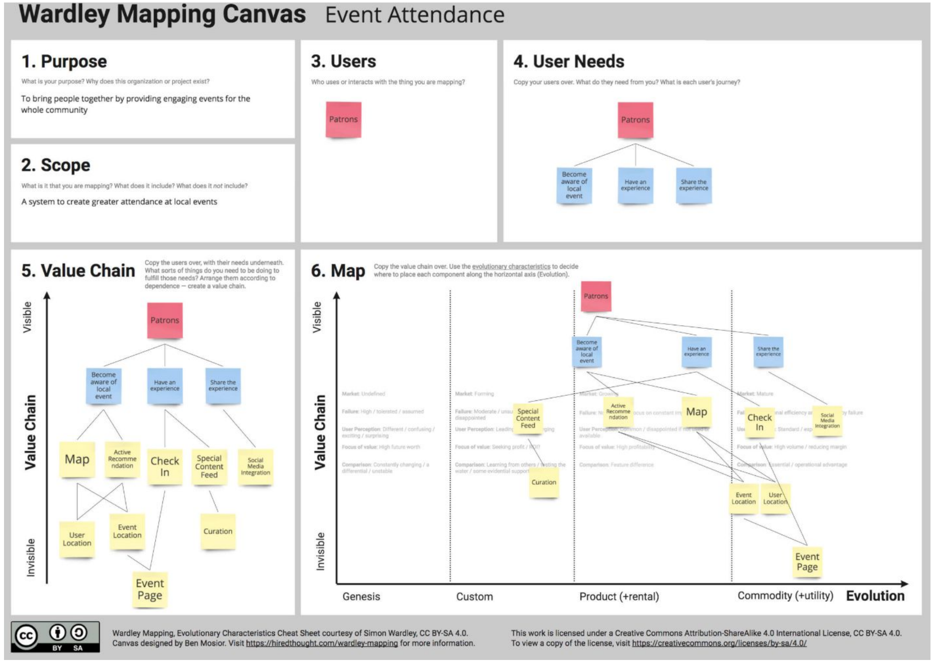
Wardley Value Map

Wardley Maps give you a visual method for talking about and developing strategy.

Wardley Mapping is all about context — increasing situational awareness.

It gives you a visual way to represent your value chain in relation to the evolution of each software component.

You end with an **understanding of things your users need** and whether the components that meet those needs are **mature or nascent in the market**


Wardley Mapping Canvas
Event Attendance
1. Purpose
What is your purpose? Why does this organization or project exist?
To bring people together by providing engaging events for the whole community
2. Scope
What is it that you are mapping? What does it include? What does it not include?
A system to create greater attendance at local events
3. Users
Who uses or interacts with the thing you are mapping?
Patrons
4. User Needs
Copy your users over. What do they need from you? What is each user's journey?
Patrons
Become aware of local event
Have an experience
Share the experience
5. Value Chain
Copy the users over, with their needs underneath. What sorts of things do you need to be doing to fulfill those needs? Arrange them according to dependence — create a value chain.
Visible
Value Chain
Invisible
Map
Active Recommendation
Check In
Special Content Feed
Social Media Integration
User Location
Event Location
Curation
Event Page
6. Map
Copy the value chain over. Use the evolutionary characteristics to decide where to place each component along the horizontal axis (Evolution).
Genesis
Custom
Product (+rental)
Commodity (+utility)
Evolution
Wardley Mapping, Evolutionary Characteristics Cheat Sheet courtesy of Simon Wardley, CC BY-SA 4.0. Canvas designed by Ben Mosior. Visit https://hiredthought.com/wardley-mapping for more information.
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License, CC BY-SA 4.0. To view a copy of the license, visit https://creativecommons.org/licenses/by-sa/4.0/

A single product might have some components that are built and some that are bought

18F

For example:

Built: the application form

Bought: identity proofing via Experian

Purchasing things that are highly mature in the market is cost-effective

Translation: if you have a problem that has been solved many times over, don’t try to reinvent the wheel

Step 3: Assess your findings and develop some documentation

Goal: Perhaps you need more information (do some market research), or you know that you need to buy something today.

Publish your discoveries in the open and give potential vendors the information they need to do a good job.

18F

Market research tools

A targeted RFI (request for information)
Review of similar products
Talk to experts
Industry days with vendors
Good ol’ Googling

A bit more on buying or building...

For a long time the accepted wisdom has been to always buy when possible and only build when no suitable packaged solution exists in the market.

18F

There are some Off-the-shelf perks

A potentially short timeline from acquisition to usage, no long term maintenance requirements and minimal or no hosting required (if the COTS is cloud-based).

Plus, building can be risky

The Dunning-Kruger effect is where IT professionals overestimate their software selection abilities and underestimate the project effort.

But. What if you need something changed to the software you just bought?

Enter the change request

“Vendor lock-in is like buying a car with the hood welded shut.”

It can be hard to make changes to the software, and if you need things customized, it will cost you.

18F

Should you choose to build it yourself… You don’t have to build the whole thing

Modular contracting can help

By breaking up what you need to buy into many pieces, you effectively de-risk the project

18F

Also: agile project management, a different way to manage the building of software that (by design) seeks to manage risk and deliver software quickly

MORE READING

https://www.thoughtworks.com/insights/blog/buy-versus-build-shift-part-1
https://www.blackpepper.co.uk/blog/trade-off-sliders-staying-on-the-level
https://realtimeboard.com/blog/wardley-maps-whiteboard-canvas/
https://www.theguardian.com/media-network/2016/sep/26/build-outsource-not-simple-choice-external-etsy
https://18f.gsa.gov/2017/09/12/how-alaska-is-using-transparency/

GOOD EXAMPLE OF AN RFI:

Boston's Smart Cities RFI
18F RFI

SOME IDEAS HERE BORROWED, thanks to 18F's:
Laura Gerhardt
Uchenna Moka-Solana