Employability in Programme Development: Establishing a labour market to higher education feedback loop drawing on local labour market intelligence

ERASMUS+: Employability in Programme Development (EPD) Project

Erasmus+ Programme, Key Action 2: Strategic Partnerships (agreement number 2020-1-UK01-KA203-079171)

Intellectual Output: IO3

Intellectual Output title: Prototype Employability Dashboard

Document title: Report reflecting on the lessons learned during the design, development and feedback stages

Contributing authors: [||]Johan Loeckx, [||]Corneliu Cofaru

Author affiliation: [||]Vrije Universiteit Brussel (VUB)

Co-funded by the Erasmus+ Programme of the European Union

report reflecting on the lessons learned during the design, development and feedback stages

Johan Loeckx, Corneliu Cofaru
Artificial Intelligence Laboratory
Vrije Universiteit Brussel

Introduction

Much valuable learning has occurred during the definition phase, organisation of the hackathons, the implementation of the employability dashboards and the surrounding discussions. We will summarize the most important take-aways for others to learn from.

1. Complementary problems arise when creating technical solutions

Technology can deblock solutions where traditional “human-centric” approaches fall short. However, they never operate in isolation and often, many *other* problems need to be solved as well for the technology – in this case the employability dashboards – to “shine”.

In our case the following "complementary” problems were identified:

1.1. *Lack of a shared vocabulary*

*Many discussion*s were difficult because the field of employability is not yet well-established, and many different terms were used for the same thing or the same term for different phenomena, both within institutions and between different stakeholders.

1.2. *Lack of a shared framework, vision or philosophy*

Furthermore, a shared vision, or even philosophy was lacking among stakeholders. Rather, they often took a quite shallow stance in which others’ views and concerns were acknowledged, but not integrated in the formulation of the problem. The academic literature as well is fragmented, and a lot of knowledge is available in the heads of career service professionals but is never published.

1.3. *Data-driven solutions need data governance*

*D*ue to the lack of a common framework to approach and discuss employability, data sources are not harmonized. Furthermore, the field suffers from poor data governance. Data are collected differently per institution, per region, with different levels of granularity and data schemas and semantics, siloed within organisations. This makes it an almost impossible task to integrate these sources.

1.4. *Methodological complexity of the task-at-hand*

The concept of employability itself is complicated, with many interacting factors. This makes it challenging to measure impact and come to “hard”, quantifiable results. Furthermore, the interdisciplinary nature makes mutual understanding difficult.

1.5. *Resistance and/or ignorance to innovation*

Finally, computerization and the introduction of AI leads sometimes to resistance, and almost always ignorance towards artificial intelligence. For this reason, quite some time was needed to introduce the topic of AI and explain its mechanisms.

1.6. *The gap between academia and industry needs time to bridge*

Overall, I have the impression that industry is very interested in the opinion and viewpoint of academia, and vice versa. However, both partners need time to adapt to each other's approaches and levels of abstraction.
Above issues had to be solved or at least “bypassed” or “short-tracked” to come to the formulation of a *problem* to be solved in the hackathon, let alone a solution.

Above issues had to be solved or at least “bypassed” or “short-tracked” to come to the formulation of a *problem* to be solved in the hackathon, let alone a solution.

2. Employability stakeholders are hard to involve

It proved very challenging to involve employability stakeholders (employers, curriculum designers, career specialists) in hackathons. We identified the following reasons:

1. Hackathons are intensive and take considerable *time* (>2 hours)
2. The output is not well-defined beforehand and thus requires a *leap-of-faith*
3. The focus is not purely on the problem of *one* stakeholder, but on a shared problem. This requires some “vision” from the participants to look beyond urgent problems, adopt a research mindset and join forces to think of future solutions.
4. Some stakeholders struggle to generalize a problem or solution beyond their local context.

3. Documented hackathon methods are of limited use

Despite the typical enthusiasm related to hackathons, we identified some gaps in the literature available in describing how to run hackathons. Mainly the role of the “mentors” was left open, as well as how to cope with heterogenous audiences. This was a particular challenge in our case, because the domain itself is ill-defined, and a shared vocabulary did not exist.

4. Problems and solutions depend heavily on the local context

The European dimension gives extra depth to the discussions on employability. However, we noticed that the contexts vary so heavily (from demographic situation, needs, structure of the economy to policies, attitudes, skills) that problems and solutions cannot be transferred.

5. It is unreasonable to expect participants to come up with AI-based solutions

The original plan was to do the matchmaking with an appropriate AI technique that solves the problem, during the hackathon. This was too optimistic and proved a distraction. It is best to keep people in their “optimal operating point”, that is, close to their field of expertise where they are comfortable.

This is even more true for hackathons with external stakeholders, as “breaking free” and being truly creative often needs some level of trust that is hard to build within a timeframe of 1-2 days.

6. “Lead-in" sessions were a necessity

Above issues made it impossible to “dive” right into a hackathon. Preparatory sessions were needed to:

1. Align on a *high-level approach*
2. Converge on the *most-important problem* to solve;
3. *Scope the problem down* to a complexity that is “manageable”, I.e. implementable within the planned time and effort