Abstract—The increased importance represented by open-source and crowd-sourced software developers and software development in general, inspired us to consider the following dilemma: can we “compute” virtuous software developers? The D-Index is our preliminary attempt. Essentially, the D-Index meaningfully equates several indicators for the virtues of a developer, such as, contributed code, its quality, mentoring in online learning communities, community engagement. Our preliminary evaluation of the index suggests that establishing the virtues for certain developers eases the identification of software “evangelists”, key success enablers for software communities.

I. INTRODUCTION

Most software currently on the market relies heavily on open-source components [1]. What’s more, the importance of open-source communities’ success is bound to increase, e.g., since many governments are leaning towards using open-source software for their own governmental infrastructures [1].

But what are the characteristics, social, organisational, technical or otherwise, that make software communities into successful software commons?

We may well hypothesise the following: “the continued success of open-source communities rests on the shoulders of bold professionals, with a multitude of skills spanning from community engagement to mentorship to coding and problem-solving; these professionals are software evangelists in that, they weigh strongly in the success of a software community and are capable of adapting its social, organisational and technical characteristics towards success”.

In lure of this hypothesis we formulated the following research question: can we “compute” virtuous software developers using data from open-source communities?

This paper discusses the D-Index, our idea of a Developers’ Virtues Index, i.e., an index of the virtues (and fallacies) of software professionals. Our ultimate goal is to understand the social, organisational and technical “skills recipe” that drives certain open-source communities to ultimate success. Assuming such a recipe exists, we hypothesis that this recipe can be elicited by isolating and studying those individuals whose key effort makes the difference in the social, organisational and technical aspects of open-source communities.

Essentially, the D-Index evaluates together the following: (a) a combination of social (e.g., communication abilities), organisational (e.g., cooperation abilities) and technical (e.g., code-quality, reopened issues) measurements for people across an open-source community; (b) a sampling of such measurements over time; (c) a comparison of observed measurements with a measurement of produced software quality. The purpose of the D-Index is to discover software evangelists with a number of possible benefits: (a) highlighting their role as community shepherds in their own right; (b) allowing researchers to study their profile, e.g., to fine-tune university training programmes; (c) engineering forges that better support and foster their existence and engagement.

Our results indicate that top open-source communities have at least one professional whose D-Index, in its current version, is above a certain threshold. This confirms that a correlation exists between successful software communities and professionals with a certain set of social, organisational and technical characteristics. We made two significant observations.

First, the correlation between open-source communities’ success and the presence of professional software evangelists with certain observable/measurable characteristics should be further investigated, e.g., to foster successful development and operations across open- and closed-source communities alike. For example, this very correlation in open-source may be a latent cause behind nasty phenomenons such as organisational rebellion [2], community forking [3] or dead open-source projects, a.k.a. vaporware [4]. Similarly, this correlation could be the reason why certain closed-source projects are driven to successful operation against strong opposing forces.

Second, more research is needed for a generalisable definition of the D-Index. In the preliminary study discussed in this paper we observed many social (e.g., ability to lend a helping hand), technical (e.g., standards for coding used or ethics for coding) and organisational (e.g., frequency of contribution or engagement) dimensions that could indeed play a relevant role in defining the virtues of software professionals but are currently out of our D-Index definition. Also, many of such variables do not have quantifiable measurements, yet. One such example is self-organization, a key characteristic of successful open-source communities. Nevertheless, our results are encouraging and seem to reflect an idea worthy of exploration.
The rest of this paper is structured as follows: Section II outlines the D-Index; Section III provides outlines the preliminary tool-support we designed to apply the D-Index in practice with a case-study (Section IV) while Section V discusses results of the practical application and elaborates on our observations. Moreover, Section VI explores a few key research efforts related to our own. Finally, Section VII concludes the paper discussing approach limitations and elaborating a tentative research roadmap.

II. TOWARDS THE D-INDEX

To engineer primordial version of the D-Index we followed four key constraints:

1) “Ranking is Based on Heterogeneous Resources": Many open-source communities provide multiple insights and measurements on the technical, organisational and sometimes social characteristics of people and projects/communities hosted within the boundary of their platforms. However, very seldom such statistics and characteristics are computed from multiple heterogeneous sources outside such platforms, e.g., representing developers’ mentoring skills (think, for example, Stackoverflow1) as opposed to their quality coding skills (think, for example, Re-Opened issues [5]).

2) “Ranking is Based on Good and Bad Characteristics alike": Many open-source platforms evaluate some forms of reputation scores based on quantity of contributions rather than quality. Conversely, our effort was concentrated on computing a reliable indicator of developers’ quality characteristics as well evaluating the magnitude of their contributions.

3) “Ranking Includes a Degradation Factor": characteristics such as liveliness and engagement are extremely important in open-source software contributors. In this vein, older contributions (either technical such as coding or social, such as mentorship through e-learning) weigh less in computing the D-Index. Essentially, the D-Index is computed by factoring every measured characteristic with its “age”.

4) “Ranking is normalised": from the pilot study conducted to arrive at a preliminary D-Index formula we noticed that some instances of the D-Index resulted into arbitrarily high numbers due to many inconsistencies and subtleties in the data sources considered (Github and Stackoverflow). Consequently, we normalised D-Index output to 0...1. Although very preliminary, this allowed us to make an evaluation of the D-Index. Further research is needed to understand the inconsistencies of the multiple possible data sources and fine-tune the formula accordingly.

Revolving around the above constraints we used focus groups and brainstorming sessions with students in addition to pilot studies on randomised open-source repositories to generate a preliminary formulation of the D-Index. The pilot study to generate a preliminary formulation was based on 2000 Stackoverlow users ratings, and 400 GitHub users as well as 150-400 accounts from randomised GitHub commits.

In its current formulation, the D-Index is defined as follows:

\[ D - \text{Index} = \sum_{\text{sources}} D - \text{Index}_{\text{NormSource}} \]  

\[ D - \text{Index}_{\text{NormSource}} = \frac{D - \text{Index}_{\text{Source}}}{D - \text{Index}_{\text{MaxForSource}}} \]

Where D-IndexNormSource is an array of (currently) two values, D − IndexSO for Stackoverlow (SO) and D − IndexGH for GitHub (GH) respectively, as follows:

\[ D - \text{Index}_{\text{SO}} = \sum_{\text{BestAnswers}} \frac{\text{BATime} \times \text{AverageBAScore}}{} \]  

\[ D - \text{Index}_{\text{GH}} = \sum_{\text{Repos}} \frac{D - \text{Index}_{\text{LastCommitTimestamp}}}{\text{watchersCount}} + \text{forksCount} + \text{stargazersCount} \]

It should be noted that D − IndexNormSource depends on how many sources are selected for measurement (e.g., we selected two sources of measurement in our case, namely, Stackoverlow and GitHub). Also, depending on the type of source formulae (i.e., formulae marked with (3) and (4) respectively) might change considerably. More research is needed to potentially generalise the D-Index formulation above to welcome multiple heterogeneous sources. Finally, more research should be invested in determining which sets of sources represent a “functionally complete set", that is, a set of sources that together exhaustively represent the skills and capabilities for software developers.

III. THE D-INDEX, A TOOL-SUPPORT PROTOTYPE

In order to test our preliminary definition of a D-Index we generated very simple tool-support using Python scripting to devise a repository crawler. The goal of the crawler is to analyse lists of typed inputs (open-source repositories such as GitHub as opposed to online learning communities such as Stackoverlow) and compute D-Index. This tool support is openly available online2.

Currently our tool-support only supports the analysis of open-source repositories on GitHub and, in parallel, mentorship contributions on Stackoverlow. To do so, the algorithm divides the computation of the D-Index in two sequential steps: first, the algorithm determines D-IndexGitHub i.e., the values of the index that refer to open-source software contributions by developers; then, the algorithm sorts GitHub communities

1stackoverlow.com/

2https://github.com/n43jl/mdpcrawl/
TABLE I
APPLYING THE D-INDEX IN PRACTICE, AN OVERVIEW.

<table>
<thead>
<tr>
<th>User</th>
<th>Reputation Stackoverflow</th>
<th>D-Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jon Skeet</td>
<td>1</td>
<td>10001.9</td>
</tr>
<tr>
<td>Baluscio</td>
<td>3</td>
<td>3508.06</td>
</tr>
<tr>
<td>Darin Dimitrov</td>
<td>2</td>
<td>3402.2</td>
</tr>
<tr>
<td>Marc Gravell</td>
<td>4</td>
<td>3039.05</td>
</tr>
<tr>
<td>VonC</td>
<td>6</td>
<td>2133.8</td>
</tr>
<tr>
<td>SLaks</td>
<td>8</td>
<td>2028.087</td>
</tr>
<tr>
<td>Greg Hewgill</td>
<td>9</td>
<td>1875.18</td>
</tr>
<tr>
<td>Hans Passant</td>
<td>5</td>
<td>1871.869</td>
</tr>
<tr>
<td>Eric Lippert</td>
<td>22</td>
<td>1814.59</td>
</tr>
<tr>
<td>Nick Craver</td>
<td>18</td>
<td>1699.23</td>
</tr>
</tbody>
</table>

members based on their quality, and filtering out repositories below the top-ten; finally, determines D-Index_{Stackoverflow}, i.e., the values of the index that refer to online mentorship contributions by elicited developers.

More in particular, the algorithm we implemented is very simple and operates as follows:

1) Find out top-users on GitHub - this ranking is public\(^3\);
2) Sort repositories and contributions by top-users using D-Index_{sorting}:
   "open-issues-count+forks-count+watchers-count+stargazers-count" - Every characteristic used in the sorting is easily computable with GitHub API calls;
3) Evaluate D-Index_{GitHub}, i.e., all values in D-Index that relate to open-source communities;
4) Search GitHub user login (or alternatively, username) on Stackoverflow;
5) Evaluate D-Index_{Stackoverflow} for developers found;
6) Sum the two D-Index values;

Finally, our tool-support includes basic mechanisms to overcome query limitations to Stackoverflow and GitHub as well as mechanisms to overcome false-positives.

IV. D-INDEX, A CASE-STUDY

To provide a preliminary evaluation of the D-Index in practice we applied it on an entire dataset collected from GitHub and Stackoverflow, excluding professionals and repositories that were previously used for our pilot study.

As previously stated, on one hand, we considered GitHub as a widely known source of quality open-source communities - from this source we elicited data necessary to correlate the quality of open-source repositories with developers and the quantity of their contribution.

On the other hand, we considered Stackoverflow as a widely known free-lance professional mentorship forum - from this source we elicited data necessary to correlate the quality of open-source developers with the quality and quantity of their mentorship contributions.

Results are contained in Table I.

The Table illustrates top-ten GitHub developers (column 1). In addition we report the developer’s rank on Stackoverflow (column 2). Finally Table I contains our evaluation of the D-Index for considered top GitHub developers (column 3). The entries on Table I are ordered by descending D-Index value.

V. DISCUSSION

In analysing our preliminary results we made three observations.

First, Table I shows top-ten D-indexing (top) GitHub users. These top-ten D-indexing professionals have score an average of 2000 on the D-Index with a peak of over 10000 for the top user, who seems to be an outlier. More investigation is needed, e.g., perhaps considering a larger number of data-points from our sources, to acquire a clearer understanding of the D-Index ranking, its validity and implications.

Second, there seems to be a remarkable similarity between the popularity ranking used by Stackoverflow (which exploits a private and closed-source algorithm) and our D-Index ranking. This could indicate that our D-Index uses metrics similar to Stackoverflow augmented with an evaluation and measurements of evidence from GitHub. This remarkable similarity suggests our approach has promise, moving in a similar direction to that of an established and widely known online software development mentorship community such as Stackoverflow. Nevertheless, computed D-Indexes could be influenced by the contributions on GitHub. More research is needed to investigate the current influences for the D-Index, e.g., to find communities in which specialised roles might influence D-Index ranking.

Third, every community has its own internal characteristics and peculiarities and consequently, in its current version, the D-Index cannot be trivially generalised. While defining and applying the D-Index in practice we learned that Stackoverflow has a very different organisational and social structure for communities and forums hosted within its platform than GitHub. In its current formulation the D-Index embeds the characteristics and peculiarities of the two communities we considered. Other communities might have even more peculiar structures. We made a similar observation while evaluating possible alternatives for a case-study. This means that further research in the direction of the D-Index should start by studying the organisational and social structures [7] typical of forges and learning communities to be considered with the D-Index, e.g., to identify formulae patterns suitable for generalisability. The focus of this preliminary study should be to determine a complete set of characteristics to be measured on many types of possible software communities (e.g., e-learning, mentorship, debugging, tech-support, IDE refinement, etc.).

VI. RELATED WORK

Although the ideas behind the D-Index are new to the best of our knowledge, a few works already elaborated on concepts related to those elaborated in this paper.

For example, works such as [8] by Morales-Ramirez et Al. combine various types of data available in software repositories such as mailing-lists to characterise software developers. The perspective in [8] is that of requirements engineering and
the goal is finding and characterising expertise levels. We offer a similar characterisation with at least one key difference: our characterisation does not reflect technical expertise in particular, rather it reflects the virtues and fallacies for the indexed development professional. Nevertheless, our work can be thought as a next step in generalising approaches such as that in [8].

Moreover, works such as [9] by Vergne and Susi apply rigorous mathematical approaches such as Markov networks to determine topics-expertise estimations in open-source discussions. Again, a key difference with our work exists: we propose the combination of approaches similar to that in [8] and [9] to study the skills profile that makes certain software developers, key performance indicators and success factors for their community.

VII. Conclusion and Research Roadmap

This paper introduces and discusses the D-Index, a first step towards determining an analytical measurement for the virtues (and fallacies) of open-source software developers. Our goal is to possibly profile the alchemy of organisational, social and technical contribution that certain key software developers bring into successful communities with their participation. This paper illustrates how the D-Index notion was realised into a workable metric and applied initially on top open-source communities. Our results indicate that evaluating the D-Index is possible and there is a strong correlation between open-source community quality and the existence of “high D-Indexing” professionals.

However, from our test trials we understood that, in its current form, the notion of the D-Index presents many limitations concerning, for example, the organisational, social and technical characteristics measured for its computation. Also, the current D-Index definition refers to no systematic metrical construction mechanism (e.g., GQM [10]) to ensure its soundness. Finally, despite our empirical observation of open-source communities in a preliminary application of the D-Index, it was not possible for us to verify the “representation condition” [11] for the D-Index, this is an essential validation of the D-Index’ soundness and generalisability beyond the context of this paper.

In light of the above premises and limitations we propose the following research roadmap.

First, more research is required to include refined organisational, social and technical metrics together. The goal behind such research should be to determine which quantifiable characteristics (social, technical, organisational) contribute towards community success. This research should start by studying empirical research into the quality of open-source communities and measurable quantities thereto.

Second, starting from the pilot study illustrated in this paper, more research should be invested in determining the D-Index’ factors in a more systematic way. This process should start by investigating the feasibility of known and confirmed structured-reasoning approaches for metrics construction.

Third, more systematic empirical research should be invested in validating the D-Index, e.g., using input form multiple input sources of multiple kind (e.g., online learning communities, MOOCs, online mentorship platforms, etc.). By all means, such research should start by focusing on demonstrating the representation condition [11] for the D-Index.

Fourth, the D-Index should be put to use in empirical studies aimed at eliciting the developers’ profiles that with a high D-Indexing. The elicited profile could then be put to good use, e.g., to refine open-source forges in support of their existence. Such multidisciplinary research could rely on Social-Networks’ analysis in combination with D-Index measurements to relate the quality of organizational/social structures observable in open-source communities with developers scoring high on D-Index.

References