19 June 2018

Code Depot Business Plan

This is a rough business plan for Code Depot, based on my current understanding of your vision, and some of my own. Nothing here is fixed; it is all open to rewriting, rewording, reordering etc.

Goals

*(The end result toward which all actions are directed)*

To create a totally new software market and ecosystem which disrupts and replaces the status quo.

More specifically worded: To transform the software market and ecosystem from its current state (consisting mainly of commercial software which is expensive, inflexible, buggy, badly supported, monolithic, invasive, insecure and untrustworthy) into one which provides software that is inexpensive, flexible, modifiable and componentized (think legos), and which is created by a loose, open (inclusive), grass-roots ecosystem of programmers and other content contributors, working on their own (or in groups), by responding directly to the needs and wants of the users.

Purposes

*(The lesser goals or route chosen to achieve the goals)*

To have an environment and ecosystem where programmers and related content contributors (including ourselves) can write open source software and be financially rewarded for it, to make a living or for extra money, on whatever they want, whenever they want and from wherever they want.

To have an ever-growing and ever-improving repository of well supported open source software components that can be inexpensively downloaded and/or used online as web services which are a (much better) alternative to the current state-of-the-art in commercially available software.

To have a profitable web-based software related business that provides a good living and any additional funds needed without requiring a fixed schedule or specific location to work from, and to do so by growing it from nothing without any need for outside capital investment.

Policy

*(Rules and standards necessary to support the goals and purposes)*

Reward actual production and support of the useful content which is chosen by and used by the community.

Any contribution, even involvement in forums or posting links to the site is rewarded.

Even the management of the ecosystem (monitoring statistics, soliciting for contributions, handling disputes etc. can eventually be handled by the ecosystem itself by incentivizing it.

Obviously, no malware or “self serving” software or mechanisms are allowed in the repository. The ultimate purpose of software is to assist users in obtaining their needs and wants, not to serve the provider of the software, or even the ecosystem, at the expense of the user.

All software in the repository is to be in the form of components which communicate with other components via flow based programming (<https://en.wikipedia.org/wiki/Flow-based_programming>).

Components can be written in any language supported by the ecosystem.

Standardized “boiler plate” interfaces are used in each component with the inputs and outputs of each component clearly defined in a standard way using metadata that allows the components to be cataloged, measured and tested by the system.

Every component submitted to the repository must be submitted with unit tests sufficient to verify that the component is operational, and also give concrete demonstrations of its use.

New policy is added, in coordination with the ecosystem (especially the stake holders), according to procedures yet to be defined.

Pricing will be worked out in coordination with the ecosystem itself, but may contain some or all of the following elements All fees may be payable in USD or CodeCache, and could possibly be “locked in”, “floating” or incremental.

* Subscription fees (locked in or floating).
* Component fees.
* Lifetime subscriptions.
* Lifetime component fees.
* Component usage (service) fee (using micropayments)
* Free trials
* Free up to a certain amount of usage or number of components.

Plans

*(How we accomplish the goals and purposes in accordance with policy)*

**Create the Code Depot Ecosystem**

Listed here are a number of programs that will be used to accomplish the plan, in the general order that they need to occur. In some cases, however, these programs may need to run concurrently. For example, some market research will need to be done before completing the book, and possibly the token engineering will need to be worked out to some degree before the market research is completed, in case we want to pay tokens as an incentive for survey or interview participation, etc.

1. **White Paper** - Complete the “Modern Ecosystem” White Paper, so it is ready to be released.
2. **Market Research** - Survey the market (open source software developers, software developers, users, content providers, potential programmers and content providers) to make sure we understand their needs and wants, their culture, their likes and dislikes, loves and hates, and so we can communicate on the right wavelength, using the right words, concepts and images.
3. **Book** - Write and publish the book that completes the White Paper.
4. **Initial Website** - Build out a website (Code Depot) that is, at first, a meeting place for programmers, content providers and users to discuss, work out the details of, and build the rest of the required functionality to make this whole thing a reality.
5. **CodeCache Token** - Engineer the CodeCache token with the correct incentives, and whatever else this entails.
6. **Launch** - Launch the White Paper, Book, website and coin all at once, in a big way.
7. **Bootstrap** - Continue to build out the website as a co-working space, repository, “store”, and whatever else it needs, with the help of any willing participants, to bootstrap it into a complete hub for the ecosystem.

Programs

These are currently just rough outlines of the programs listed above under the plan. Ideally and eventually they will be step-by-step sequences.

**White Paper**

* Keep it to a high level overview.
* It’s a lead-in to the book.

**Market Research**

* Surveys (on-line, in person, formal and informal)
* Interviews (on-line, in person, formal and informal)
* Publics:
  + Open Source Programmers
  + Non-Open Source Programmers
  + Open Source Users/Enthusiasts
  + Commercial Software Users
  + Potential content contributors
  + Blockchain/Crypto enthusiasts
* Subjects/Demographics:
  + Likes/Dislikes (Software, work, computers)
  + Loves/Hates (Software, work, computers)
  + Occupation
  + Age
  + Gender
  + Political Affiliation
  + Residence
  + Income bracket
  + Education
* Participation in surveys and interviews could be incentivized with CodeCache tokens.

**Book**

* Possibly free in electronic format.
* Published in parts? Continuously?

**Initial Website**

* Initially a collaboration space.
* Bootstraps itself through contributions to the code-base into the full deal.

**CodeCache Token**

* Must incentivize properly.
* Founders get an initial payment and a small cut of every token payout.
* Mining is done by “proof of actual work/contribution to the project”. When someone contributes in a material way or his code or content is used (or however we work it out), he has mined a token.
* The pricing of the service has an effect on the value of the token. If it’s dirt cheap to subscribe to use the components or receive support, who needs the tokens? If it’s expensive, the token value goes up. If it gets too expensive, though, then commercial software/services starts to look more attractive. On the other hand, all you have to do is contribute enough feedback, code, content, promotion or help and “earn your keep”, rather than having to pay for it. There is a very interesting and dynamic set of incentives at play here.

**Launch**

* Lots of outlets at once:
  + News releases
  + Email blast
  + Banners
  + Meetups
  + Open Source forums
  + Etc.

**Bootstrap**

* Orchestrated sequence of events resulting in the full ecosystem being built out.

Statistics (Metrics)

*(the things we measure)*

* Working components added.
* Download of components.
* Content items added.
* Content items downloaded/read.
* Help requested.
* Help provided.
* Time of response to help requests.
* Result of help requests.
* Usage of components.
* Total Token “Capitalization”.
* Site hits.
* Unique hits.
* Sign-ups.
* Total user base.
* Total contributor base.
* Positive feedback.
* Negative feedback.
* Bug reports.
* Bug fixes.
* Average usage of components per user (and standard deviation)
* Average usage of content per user (and standard deviation)

Ideal Scenes

*(How things look when everything is going right. Used to recognize and investigate problems)*

The user base is rapidly and constantly expanding and giving rave reviews, bringing in their friends, giving feedback (the more positive the better, if it’s honest). Usage per user is increasing.

The contributor base is rapidly and constantly expanding, and producing more and more, better and better code and content, and providing help to users.

The perceived value of the CodeCache token is increasing, in spite of the constant inflation of the token caused by additional components, content and services being provided.

QUESTION: If tokens are mined to pay for services, doesn’t that result in too much inflation? Should the coins for services come from the existing pool somehow?

QUESTION: If components or content become outdated and go out of use, what effect does that have on the economy of the token?

Valuable Final Products

*(The valuable items that have are produced and exchanged with the world)*

Productive programmers earning a living without having to have a “job”.

Excellent software components and systems of components that are well supported and enhanced and modified as needed.

An ecosystem and internal economy that people can be a part of, talk about, contribute to and build things with.