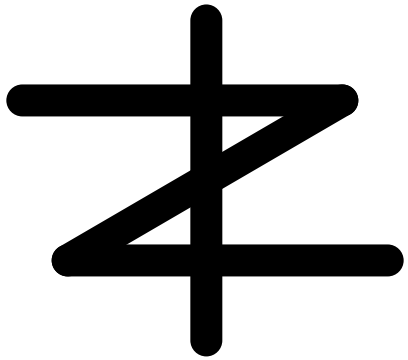
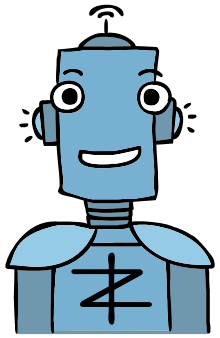


Zerocracy, Inc.

Zerocrat Key Features
www.zerocracy.com



Zerocrat is a chatbot that automates *routine* project management operations for key software project stakeholders.

The green features are implemented already, orange ones are in progress, while the red ones are still in the backlog.

Who Are the Stakeholders?

Product Owner

Funds the project, provides business requirements, accepts new releases from the Architect, approves hourly rates of Developers, reports bugs. There are usually one or two Product Owners in the project.

Architect

Boostraps the project, builds proof-of-concept, accepts bug reports and feature requests from Developers, releases new versions, regularly reports to the Product Owner. There is usually one Architect in the project, but more complex and risk-averse projects may want to assign a secondary Architect as a backup.

Developer

Implements features, fixes bugs, reports new bugs. There could be many Developers in a project, but a manageable and effective team usually include 10-30 people. Smaller teams tend to be slower than expected, while larger ones experience too many deadlocks and communication clashes.

Reviewer

Reviews pull requests and suggests corrections, gives those pull requests the green light for merging. Zerocract selects reviewers semi-randomly from the entire project pool of Developers.

Tester

Attempts to break the product, reports new bugs, accepts bug fixes from Developers. The best teams include as many Testers as they have Developers. The quality of the product only grows if the number of Testers is large.

Quality Assurance

Reviews the quality of communcation, makes sure Developers behave compliantly to the Policy, approves tickets closure. Usually, there is a single QA person in a project, while having a backup is advisable.

A more detailed analysis of project roles can be found in this blog post:
Key Roles in a Software Project.

Operations

Assigns Tasks

Zerocrat picks up a task from the entire scope of the project and decides which developer is the most suitable for it. There is a multi-factor election process explained in §3, which takes into account all visible and previously collected metrics of each developer, both in the current project and other projects in the entire platform. The election results become visible for the developer.

Resigns Tasks

Voluntarily or forcefuly Zerocrat may take a task away from a Developer and put it back to the backlog. This operation may lead to certain consequences to the person who initiated such a request.

Discovers Puzzles

Puzzle Driven Development, a patent pending method (application 12/840,306), is the core element of Zerocracy management model, which allows Developers to decompose larger pieces of work into smaller increments without integration with the management. Zerocrat, with the help of 0pdd, fully automates this process.

Requests Code Reviews

Each pull request, according to §27, has to pass the first code review of a peer Developer and the second one of an Architect. Once a pull request is submitted Zerocrat finds the most suitable code reviewer, assigns to the task, and makes sure the rewards are paid accordingly.

Chats

Zerocrat is a rich chatbot with a minimalistic web dashboard. It chats with project stakeholders via Slack, GitHub, Telegram, Jira, Trello, WhatsApp, WeChat, and over 20 other messengers and ticket tracking systems. This GitHub issue illustrates how chatbot communicates with programmers: zerocracy/farm#1282.

Registers Impediments

Some tasks may have dependencies, which have to be resolved prior to the continuation of the task at hands. Per §9 Zerocrat allows a Developer to put any task on hold for some time, to prevent it being taken away and re-assigned. Further, Zerocrat applies certain limitations to prevent the task of being held for too long and to prevent deadlocks.

Reminds

Zerocrat remembers the deadlines for each task and reminds its performers when time is over or close to that point. Zerocrat has an internal reminders management sub-system to make those reminders most effective.

Sequences Tasks

According to the priorities, inter-dependencies, and resources availability Zerocrat decides which tasks must be completed first and which ones can wait. The decisions made affect the schedule of the project and task assignments.

Labels Tasks

Zerocrat utilizes management features of the ticket tracking system, like badges, labels, priorities, assignees, projects, milestones, and so on, in order to make its integration with the human community seamless.

Rewards

Rewards a Task

When a microtask is completed Zerocrat pays the developer via PayPal, Upwork, Bitcoin, Zold, or some other payment system suitable for worldwide micro transactions. Aside from the monetary reward, the developer gets positive reputation points, which are a valuable driver of the gamification system.

Punishes for Delays

When a task is not completed in time, which is usually ten days, Zerocrat takes the task away from the developer and attempts to assign it to someone else. A certain amount of negative reputation points go to the guilty delayer, as per §8.

Penalizes Slow Reviews

Slow code reviews, while sometimes perfectly justified, often become a bottleneck for a project that runs forward. Zerocrat punishes the Architect if a slow code review is detected, as per §55.

Boosts a Budget

As per §5, an Architect is allowed to increase the fixed budget of a task a few times. Sometimes this manual "boosting" is required, especially when the task is too complex *and* impossible to break down into smaller increments. Statistically speaking, boosting is required in less than 3% of all tasks.

Pays Bonuses

There are situations when a Developer deserves a bonus, either in reputation points or monetary ones. For example, for completing a task faster than usual, as §36 explains.

Pays Extra

The Architect may instruct Zerocrat to send an arbitrary amount of cash to any Developer, when the Architect feels like this, as per §49. This feature is strongly discouraged to use, since it goes against the XDSD philosophy of money per result only. However, sometimes it's necessary to use this feature, especially in exceptional situations.

Pays For Bugs

According to §29 Zerocrat rewards everybody who manages to submit a defect, which is approved by the Architect and got in to the project scope, be it a Tester, Developer, or even a Product Owner. The logic behind this was explained in *More Bugs, Please* blog post, with a few practical examples.

Rewards a Release

Every time a new version is released, Zerocrat rewards the Architect with a bonus, calculated by a formula explained in §54. The reasoning behind the formula is to motivate the Architect to release larger pieces of functionality more frequently.

Rewards a Review

Every time a pull request is closed, the Architect gets a small bonus for the code review completed, per §28. The bonus motivates the Architect to stay on top all changes being merged.

People

Assigns a Mentor

Zerocracy is an invite-only platform, which means that in order to become a Developer one has to be invited by someone already registered with us *and* having a decent reputation score, as per §1. The inviter becomes a "mentor" who earns a certain "tuition fee" as part of everything the student is making, until the student "graduates" after reaching a high enough reputation, as explained in §43.

Screens

Every Developer has a few public profiles, including GitHub and Stackoverflow, which more or less accurately discribe the person from technical perspective. Zerocrat scrapes them time to time, parses the data, and analyse it. On top of that Zerocrat performs regular screening interviews, to understand better who is working with us.

Builds a Team

Any Developer can freely join any project, provided the project Architect accepts the joining request. However, newbies are not even allowed to apply, until they spend some time in "sandbox" projects and build a decent reputation. Zerocrat manages the entire graduation process, as explained in §2, §13, §33, and §35.

Promotes

The Product Owner can promote a project, as in §26, or an open vacation in it, as in §51, using the Board of Zerocracy and all available communication channels with registered Developers. Thus, having a project in Zerocracy the Product Owner has free access to a large pool of engineers, which completely eliminates the necessity of recruting, hiring, interviewing and so on.

Spies on the Team

Zerocrat observes the performance of the team and reports to the Architect when someone is underperforming or any other human resource management indicators fall of the expected borders.

Invites

Observing the situation in different projects Zerocrat decides which ones are in need of programmers and which Developers are most suitable for specific projects. Zerocrat regularly invites the right Developers into the right places.

Picks a Project

Any Developer can freely join any project, provided the project Architect accepts the joining request. However, newbies are not even allowed to apply, until they spend some time in "sandbox" projects and build a decent reputation. Zerocrat manages the entire graduation process, as explained in §2, §13, §33, and §35.

Defines Rates

Looking at the results delivered by Developers Zerocrat makes a decision of who deserves a raise or a decline. The decision of Zerocrat is not disputable, but doesn't affect the effective rates in each project. It affects the marketable rate of each Developer, while the local rate stays in hands of the Product Owner.

Vacations

A Developer may decide to go on vacation and ask Zerocrat to stop assigning new tasks. This process is penalty free and may be used freely by Developers in order to manage their work time as they wish.

Forecasts

Collects Metrics

Zerocrat, via a constatly growing collection of adapters, collects metrics from the platforms where the code is being developed and programmers communicate, e.g. Git, GitHub, Jira, Jenkins, SonarCube, and so on. The projects get benchmarked by the metrics collected.

Re-Estimates

Zerocract regularly requests randomly selected Developers to re-estimate the project, using the method of Scope Champions explained in the patent application US 12/193,010. Zerocrat uses provided estimates to predict the future of the project, both in terms of time and cost.

Determines Budget

Using 1) project metrics, 2) benchmarking data across all projects in the platform, 3) risks and 4) estimates provided by Developers, Zerocrat recalculates the future of the project and determines the budget, with certain accuracy and precision. The budget is recalculated multiple times a day, helping the Product Owner stay on top of the current project situation.

Develops Schedule

Similar to the budget calculation, Zerocrat builds up project schedule, using statistical information and estimates. The schedule is also delivered to the Product Owner with certain accuracy and precision.

Estimates Change Requests

Zerocrat can take a change request, with modifications to key project parameters, runs full estimate of the the budget, scope, and costs, and demonstrate the outcome.

Analyses Risks

Any project stakeholder may register a number of risks, with their key qualitative and quantitative characteristics. Zerocrat analyses the risk register and converts its fallback and mitigation plans into tasks in scope.

Identifies Risks

By looking at the current situation in the project and the history of all previous projects under management Zerocrat predicts what events may occur and what are the probability and impact of them, and registers them as risks in the project risk register.

Benchmarks

Zerocrat manages hundreds of projects at the same time and thus has direct access to the metrics in all of them. Using this data Zerocrat compares the current project with the statistical performance of others and makes recommendations to the Architect and the Product Owner to help them make preventive and responsive corrective actions.

Calibrates

Zerocrat enables calibration and fine-tuning of key performance parameters, according to the learnings obtained during the course of the project and to the input of the Product Owner and the Architect.

Snapshots

A quick summary of project affairs, also known as “snapshot,” can be delivered to an interested stakeholder by request. The snapshot includes all key indicators of project progress, including scope, cost, time, risks, quality, and staff data, and a short list of recommended corrective and preventive actions.

Process

Enforces Rules

There are multiple rules in the Policy, which have to be enforced in order to make sure the methodology works. For example, Zerocrat punishes for tasks refusal per §6, for manual task assignment per §19, task boosting per §15, and so on.

Reads the Policy

Zerocrat doesn't have any hard-coded parameters inside. Everything that is needed for its work decisions it takes form the Policy, which is published online as an HTML document and is "open source." The community of programmers actively contributes to the Policy, to make it better, more fair, and more effective.

Demands Bureaucracy

Aside from writing code the process must include a certain amount of procedures, which must be regularly completed by project stakeholders and produce reports. CMMI-Dev is a perfect example of a set of rules Zerocrat may apply to a software team. RUP and MSF are also good examples. The reports may be related to the architectural decisions made, requirements definition and acceptance, supplier management, and many other process areas.

Reviews Quality

The Quality Assurance reviews every completed micro-task and gives their "quality verdict," which affects the rewarding formula Zerocrat uses to compensate the work of Developers and code reviewers, as in §30.

Trains

Zerocrat analyses the behavior of all project stakeholders and suggests corrective and preventive actions, related to the quality of process they are supposed to follow.

Self-Reflects

Zerocrat looks back at the actions performed by people and by the robot and makes decisions about the quality of them. The decisions are then presented to Zerocracy management team for corrective actions.

The ultimate goal is to make Zerocrat fully compliant with ISO 9001 and CMMI standards.

Scaffolding

Bootstraps a Project

Each project in order to be started, as explained in §12, has to be configured, connected with the ticket tracking system, granted proper permissions, and so on. The Architect earns a commission from all funds spent in the project.

Dispatches RfPs

In order to find an Architect the Product Owner can submit a short summary of their project requirements to the "Request for Proposal" management subsystem. Zerocrat will announce the project among the most reputable Developers and one of them will pick the project up. This process is described in §40 and §41.

Funds a Project

The Product Owner may add funds to the project via credit card (we are using Stripe). According to §21 it is required for a project to have funds on board before any tasks can be assigned to anyone. Funds are released only when tasks are completed. All unused funds, in case of project termination, can be refunded to the Product Owner, as per for §22.

Refunds

Zerocrat automatically re-charges the credit card of the Product Owner when the project runs out of funds, as explained in §21. To stop that the Product Owner has to put the project on pause and then restart it.

Sends Money

Zerocrat accumulates micro payments and sends them out when the transaction amount is large enough. Zerocrat is integrated with payment processing systems via their APIs.

Pauses a Project

The Product Owner can put the entire project on hold at any moment, just by asking Zerocrat to do that, as explained in §24. The project can also be resumed any time. When the project is on pause, no funds will be spent, except for the tasks assigned before.

Exposes API

It is possible to interfact with Zerocrat via a RESTful JSON/XML API, in order to download the data, give instructions to the chatbot, and receive responses.

Reports

Zerocrat reports to the Product Owner during the course of the project, both via a chatbot and via HTML documents in the dashboard. The majority of project documents are stored in XML files and can be dowloaded "as is" from the dashboard.

Vests Equity

The Product Owner can rewards Developers not only with money, but equity of the project. Zerocrat automates this process and distributes equity in micro installments when tasks are getting completed, as explained in §37.

Consolidates

Indidividual projects may be gathered into portfolios, in order to consolidate reports, including cost, time, and scope. Portfolios not only aggregate the data, but enable exchange of resources, cross-project inter-dependencies, and so on.

Destroys a Project

The Product Owner can terminate the project and entirely remove the data from Zerocracy servers, as promised in §25.

There are over a hundred micro robots inside Zerocrat, which perform the described operations, and the number of them is growing.

Twitter Facebook Blog

Pitch Executive Summary Features

555 Bryant Str, Ste 470
Palo Alto, CA 94301
408.692.4742
team@zerocracy.com

0.14.0 January 5, 2019