How a Hackathon appreciates quality code

Some lessons from the world’s largest Blockchain hackathon

yesterday

Robin Knaapen of team DC Cicero shows off their 10 out of 10 score on code quality at the worlds largest Blockchain hackathon. Image courtesy of Drenthe College.

Last weekend, the world’s largest Blockchain hackathon took place in The Netherlands. During **54** hours, **350** participants, organised into **55** teams, competed in **five** tracks, on **one** grid, and with **zero** time to lose.

Also, these teams took responsibility for the quality of the code they were hacking together by executing **2000** code quality checks.

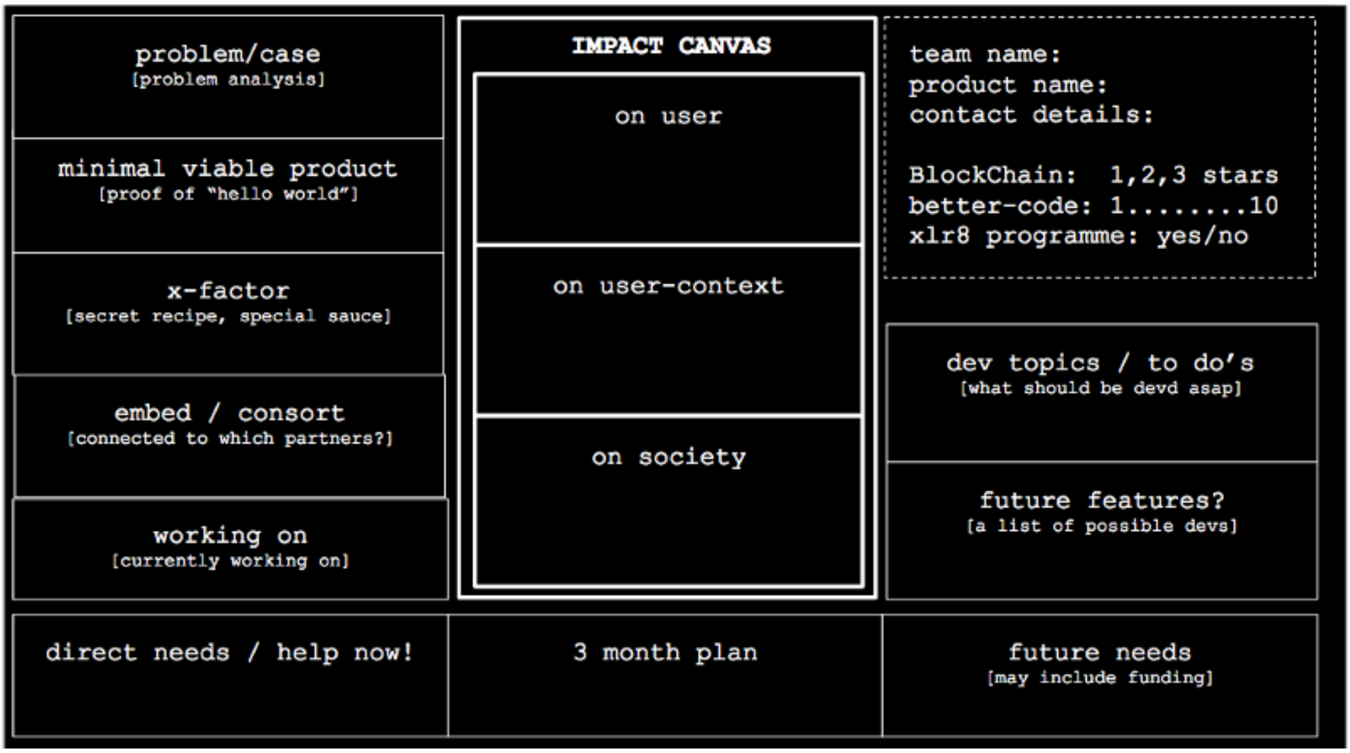
Here is how we made those 2000 checks possible, and how we made sure they helped to accelerate the teams.

An appreciative approach to judging

The hackathon organisers decided to take what they call an “appreciative” approach to the judging process. In their words:

> *we will NOT have a beauty contest and judge you afterwards based on smooth sales pitches or your shiny smile*

Rather, they facilitated a holistic approach, throughout the event and beyond, in four stages: (1) support, (2) reward, (3) award, and (4) accelerate. A key instrument in this setup is the “Impact Canvas” on which the teams share the wealth of their projects on multiple dimensions.



problem/case
[problem analysis]

minimal viable product
[proof of "hello world"]

x-factor
[secret recipe, special sauce]

embed / consort
[connected to which partners?]

working on
[currently working on]

IMPACT CANVAS

on user

on user-context

on society

team name:
product name:
contact details:

BlockChain: 1,2,3 stars
better-code: 1........10
xlr8 programme: yes/no

dev topics / to do’s
[what should be devd asap]

future features?
[a list of possible devs]

direct needs / help now!

3 month plan

future needs
[may include funding]

The Impact Canvas template. Space for sharing all dimensions of a great hackathon project. Kudos to Ritzo ten Cate for driving this approach.

Using the canvas, the teams share and discuss with supporters, Jedi, accelerators, and reporters, taking them along on their journey, engaging them, co-creating with them.

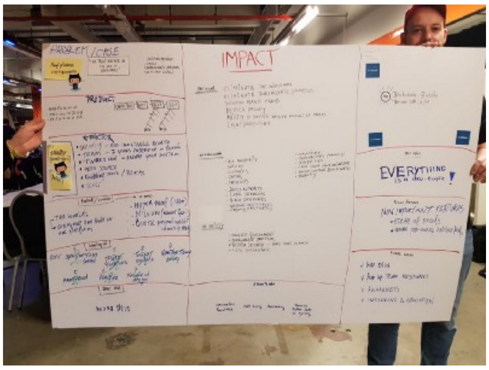
How about code quality?

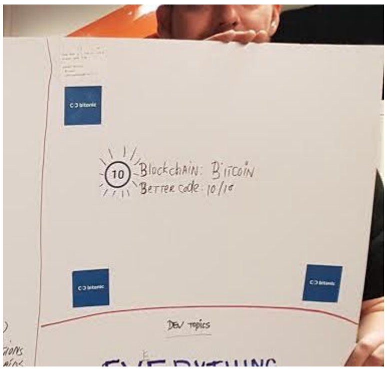
One of the dimensions of a great hackathon contribution is the quality of the solution, including the code. But how can teams collaborate on code quality and share their code quality achievements with others?

For this, we gave them the Better Code Hub:

We gave the team **10 guidelines for building maintainable software**. The guidelines were presented to them in a pre-hackathon meet-up and they received copies of our O'Reilly book *Building Maintainable Software* that describes the guidelines. Together, the 10 guidelines provide a *Definition of Done* for code quality that is challenging, yet achievable and fun.
We gave the teams **free access to the Pro version of Better Code Hub**, allowing them to check their code against the 10 guidelines and giving them actionable feedback on what changes to make in the code for meeting the guidelines. Teams can head over to www.bettercodehub.com to get feedback within minutes, and they can turn on *Push&Pull support* to get feedback on every commit.
A team of software quality Jedi were available during the event to help teams use the Better Code Hub and for **counseling** them on anything related to code quality. Which refactoring should I apply? How can I deal with dependencies on third party code? How can I make my code more testable? The code quality Jedi's have answers.
Before code-freeze on the last day of the Hackathon, all teams were given the opportunity to have their code quality results **validated** by the Jedi, who provided them with swag and badges and encouraged them to **share their code quality score** on their Impact Canvas.
After the event, the teams retain access to the Pro version of Better Code Hub **for another 12 months**, as well as continued support from the Jedi on anything related to code quality. The hackathon is a great learning experience for putting together a complete application, including third-party components and build pipeline, but it's only the first step towards a shippable product.

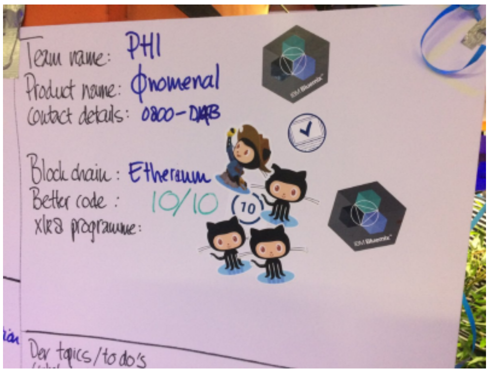
This is what code quality looks like on some of the Impact Canvases:





Blockchain: Bitcoin
Better code: 10/10

Dev topics



Team name: PHI
Product name: Φnomenal
Contact details: 0800-DAB

Block chain: Ethereum
Better code: 10/10
xlr8 programme:

Dev topics/to do's

IMPACT CANVAS

ON USER

Direct control on budget

Empower citizen

ON USER-CONTEXT

Idea generation from citizens

TEAM: GO-CHAIN

PRODUCT NAME:

SELF-GOVERNANCE SYSTEM VIA BLOCK CHAIN

CONTACT DETAILS:
HERVÉ FRANÇOIS

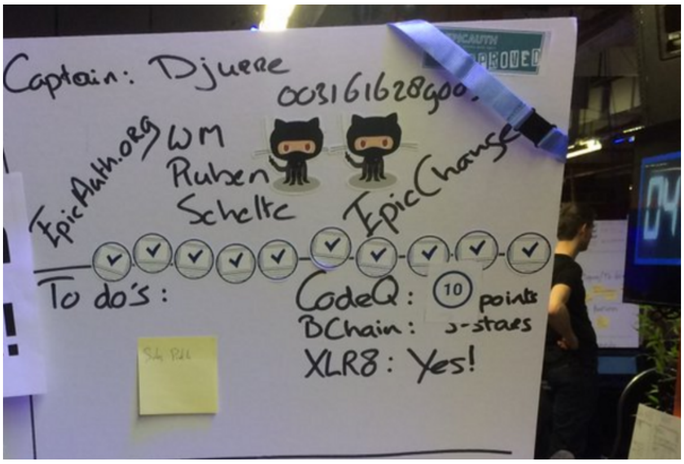
BLOCKCHAIN:

WHY BLOCKCHAIN:
- TRACK & TRACE
- SCALABILITY
- VOTERS SELECTION
- USE OF SMART CONTRACTS

BETTER- CODE: 8

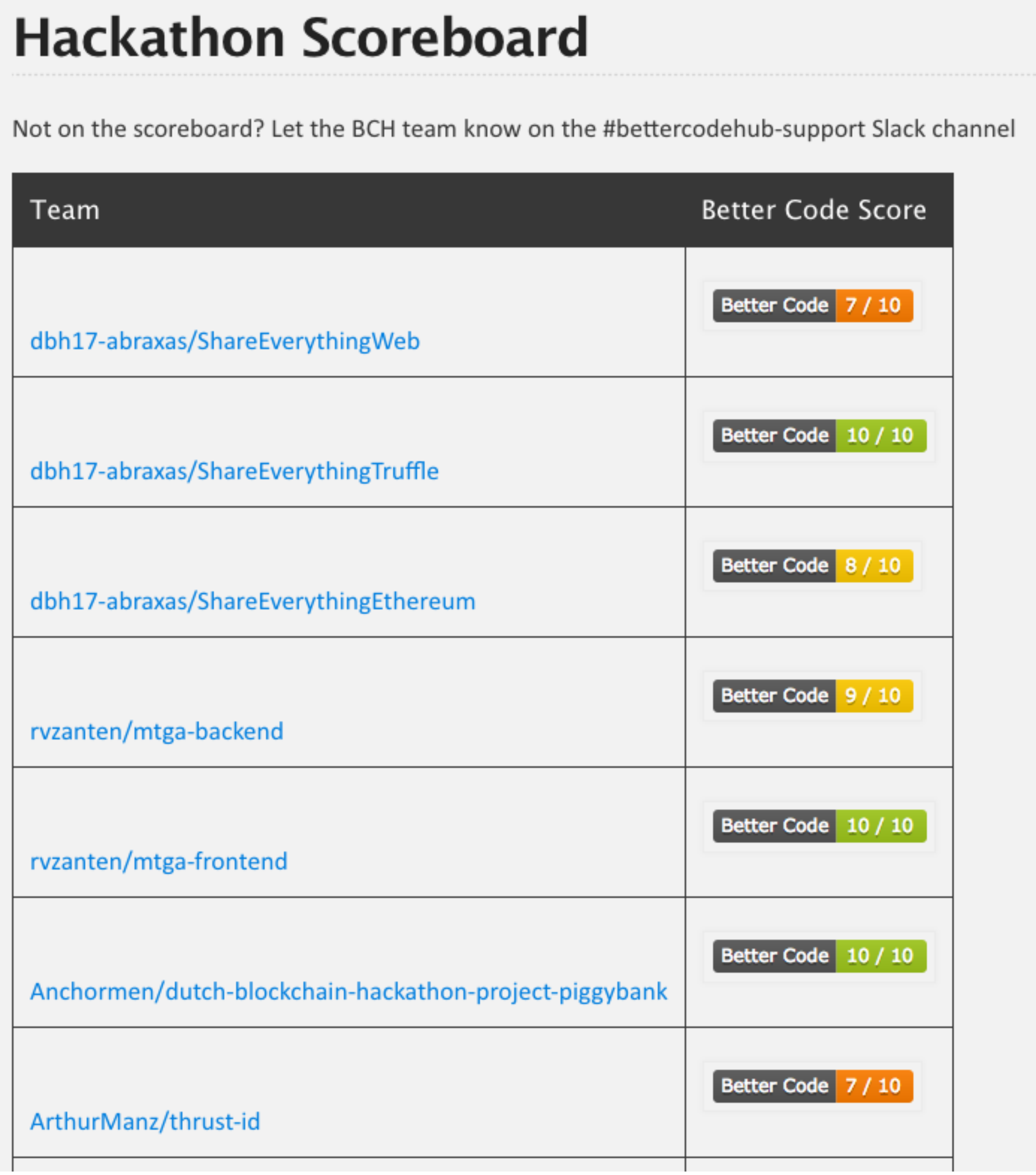
XLR8 PROGRAMME:

DEV TOPICS / TO DO'S:



Code quality achievements on the Impact Canvas of 3 teams at the Dutch Blockchain Hackathon.

Digitally, teams showed off their code quality prowess with GitHub badges that were collected on a scoreboard:



Hackathon Scoreboard

Not on the scoreboard? Let the BCH team know on the #bettercodehub-support Slack channel

Team	Better Code Score
dbh17-abraxas/ShareEverythingWeb	Better Code 7 / 10
dbh17-abraxas/ShareEverythingTruffle	Better Code 10 / 10
dbh17-abraxas/ShareEverythingEthereum	Better Code 8 / 10
rvzanten/mtga-backend	Better Code 9 / 10
rvzanten/mtga-frontend	Better Code 10 / 10
Anchormen/dutch-blockchain-hackathon-project-piggybank	Better Code 10 / 10
ArthurManz/thrust-id	Better Code 7 / 10

The Better Code Hub badges were collected on a central scoreboard.

So what is the impact of code quality during a hackathon?

We chose not to just measure code quality and feed measurement results to the jury. That would run counter to the “appreciative approach” and have the wrong effect. Instead, we empowered the participants to get code quality right, in a uniform manner, enabling them to show off their code quality achievements as one of the many dimension of their achievement. The effects are as follows:

Uniting teams: with a clear definition of done for code quality, teams don't need to waste time quibbling about coding style. Also, they get to celebrate together, each time a code check shows improvement.
Faster coding: teams that build high-quality code from the start can keep up their coding speed throughout the hackathon. Also they can more easily handover code within the team. No spaghetti to get entangled in.
Appreciation for code: Hackathons should be about code, as much as they are about learning, team work, and brilliant ideas. Making code quality easy to measure, easy to share, and easy to reward contributes to give code the visibility it deserves.

So, if you are organising a Hackathon, we can recommend an appreciative approach, including an empowering approach to code quality. If you want to learn from our experiences at the Dutch Blockchain Hackathon, we are happy to help.

Joost Visser is CTO at the Software Improvement Group, Professor of Large-scale Software Systems at Radboud University, and author of O'Reilly books "Building Maintainable Software" and "Building Software Teams".