A Trading Market to Incentivize Secure Software

Malvika Rao

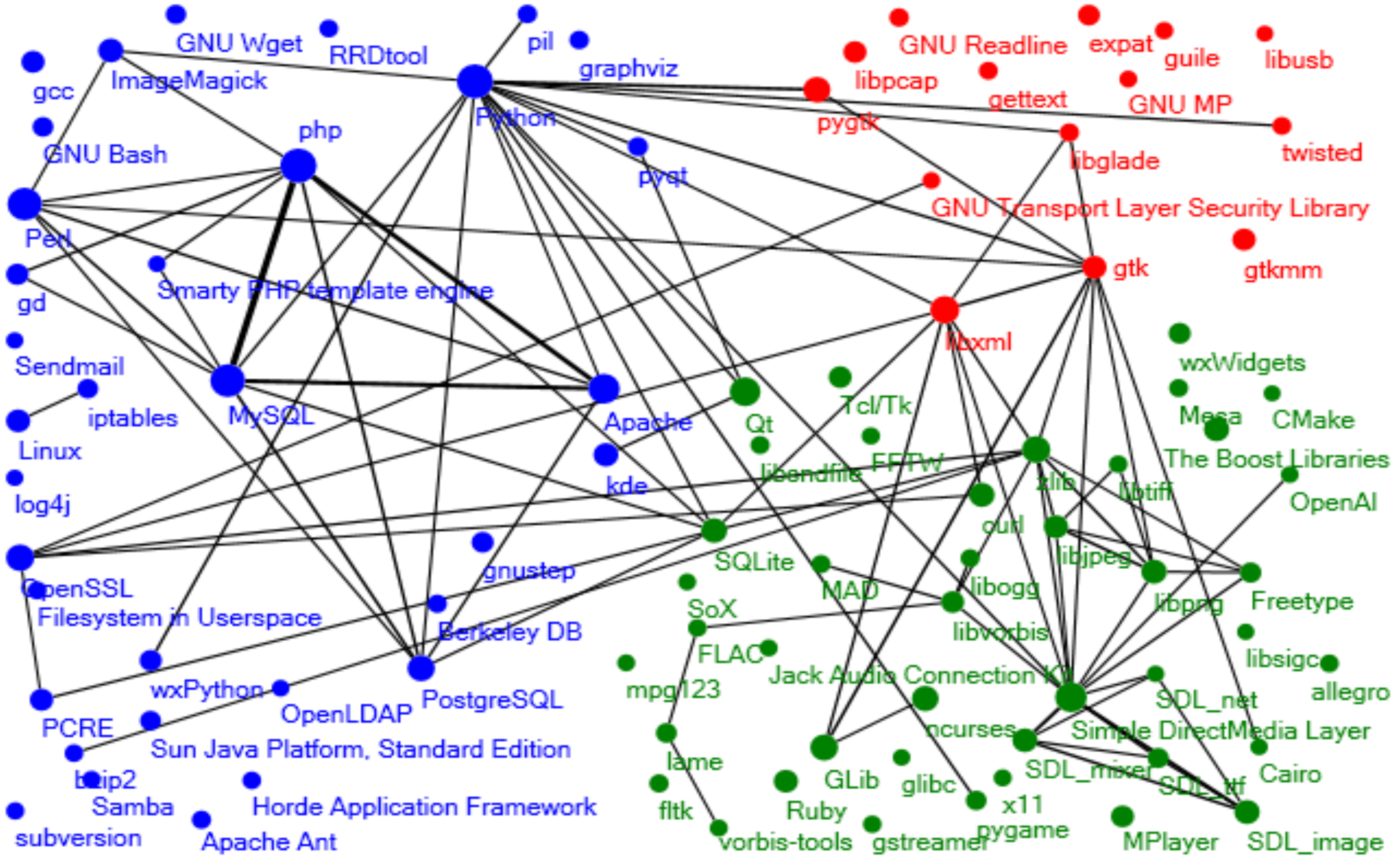
Incentives Research Inc.

Joint work with Georg Link (University of Nebraska at Omaha), Don Marti (Mozilla), Andy Leak (Mountain View Smart Contracts), Rich Bodo (Mountain View Smart Contracts)

Outline

- Introduction
- Background and motivation
- Our approach
- Simulation
- Ongoing and future work
- Conclusion

Software today



Software today

- Software systems have evolved from an "input tape to output tape" model *to one of continuous interaction with other large systems*.

- Today's software systems exhibit behavior that is decentralized, interdependent, and dynamic.

Peer Production

(Coase's Penguin, or, Linux and The Nature of the Firm, Benkler 2002)

- A form of production where individuals self-organize to collaborate, self-match to tasks best suited to them.
- Characterized by decentralized conception and execution, diverse motivations, where projects are governed as common property regimes.
- Example: FOSS

Peer Production

- What could be better?
  - Funding digital infrastructure

    (Roads and bridges: the unseen labor behind our digital infrastructure, Nadia Eghbal 2016)

  - Price signals to align supply and demand

    (Kooths et al. 2003)

Research question

How can a market design incorporate price signals into peer production, facilitate information sharing, and promote quality?

Outline

- Introduction
- Background and motivation
- Our approach
- Simulation
- Ongoing and future work
- Conclusion

Our approach

- A trading market for software issues. Prototype is called *Bugmark*.
- Participants can *trade* futures contracts to elicit information on issues and to incentivize work.
- Users, developers, testers, project maintainers, investors, and others can trade on questions such as: *will a bug be fixed? will a vulnerability be found?*

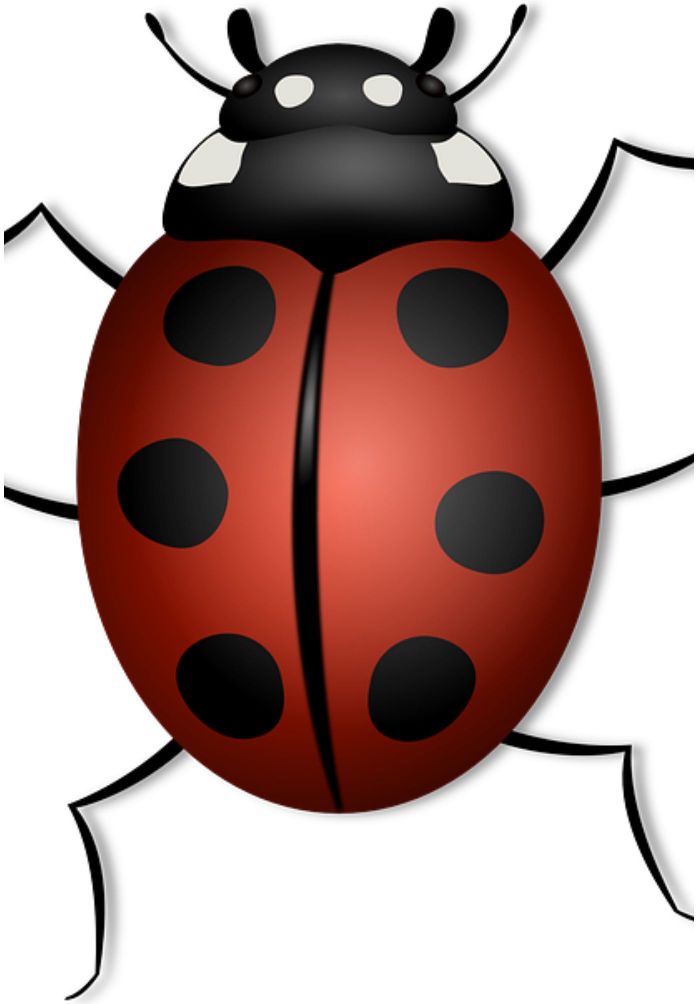
Our approach

- *How does it work?*

- User Adam finds a bug

- Adam is not a developer but has heard of Bugmark where he can pay to get the bug fixed

Courtesy Georg Link



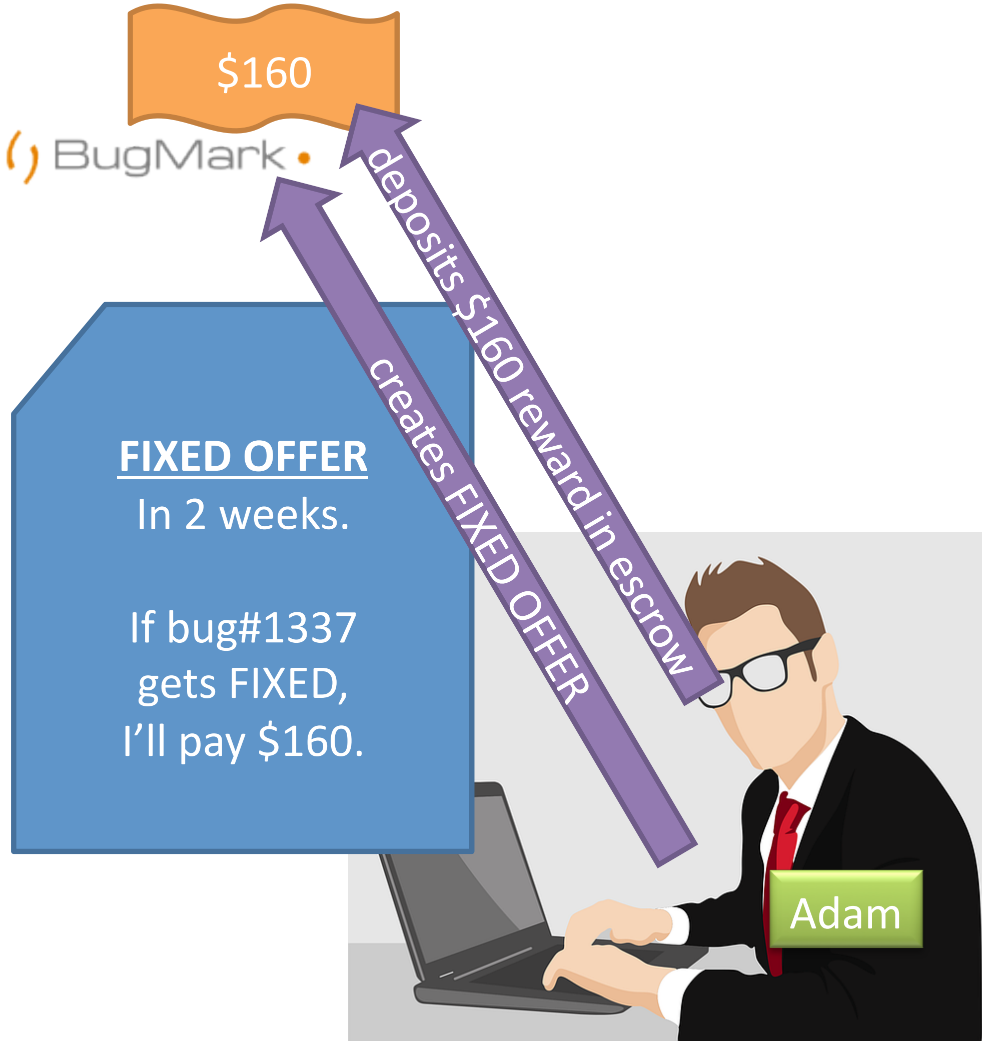
Step 1: File or find a bug

- Before Adam can go to Bugmark, he has to make sure the bug is documented.
- Bugs are documented in an issue tracker.
- If no entry exists for this bug, Adam creates one.
- Adam's bug is identified as bug#1337

Courtesy Georg Link

Step 2: Offer to pay for the bug fix

- Adam goes to Bugmark and offers to pay for the bug fix
- Adam creates a FIXED OFFER with a maturation date in 2 weeks for a payoff of $200
- Adam buys 200 units at a unit price of $0.8, paying $160



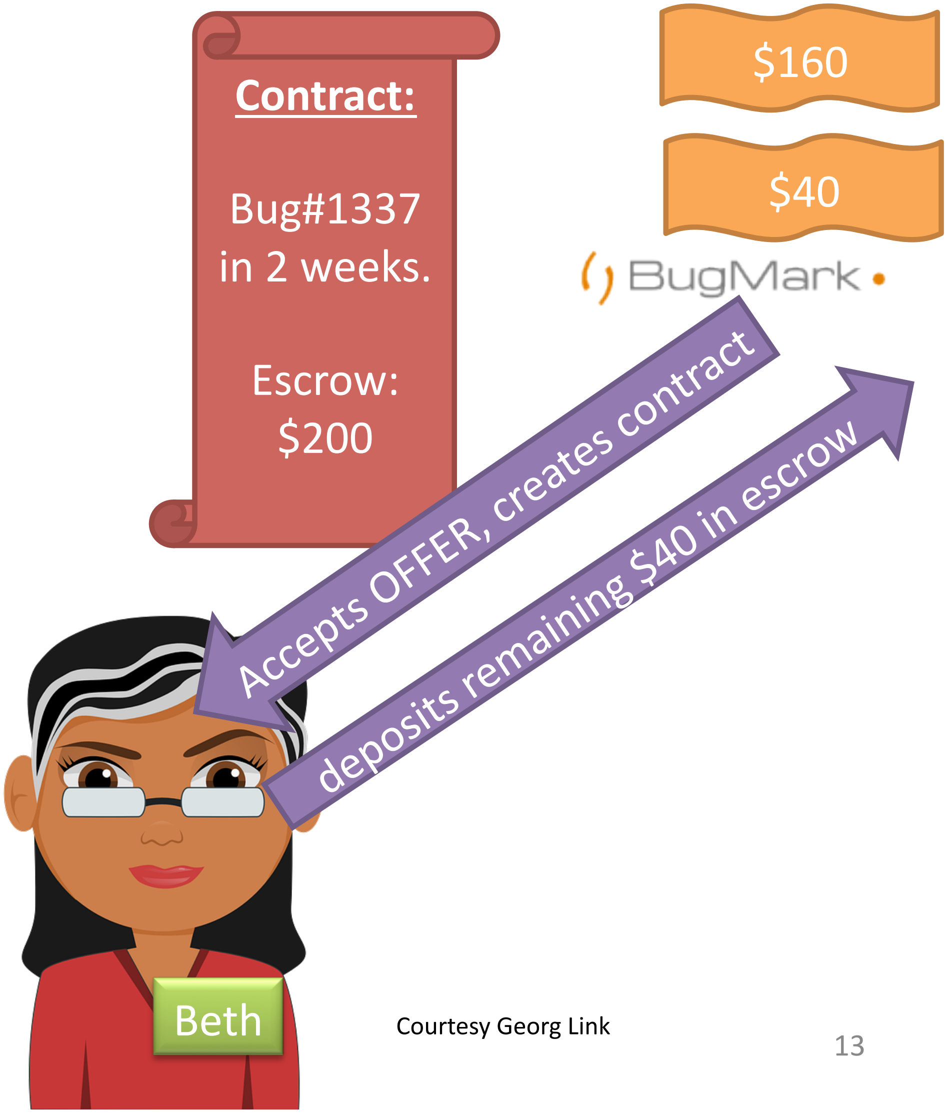
FIXED OFFER
In 2 weeks.

If bug#1337 gets FIXED, I'll pay $160.

Courtesy Georg Link

Step 3: Developer accepts FIXED OFFER

- Developer Beth sees the FIXED OFFER, has time to fix the bug within two weeks, and decides to buy the FIXED OFFER
- Beth buys 200 units at a unit price of $0.2, paying $40
- Contract is formed



Contract:

Bug#1337 in 2 weeks.

Escrow: $200

$160

$40

BugMark

Accepts OFFER, creates contract

deposits remaining $40 in escrow

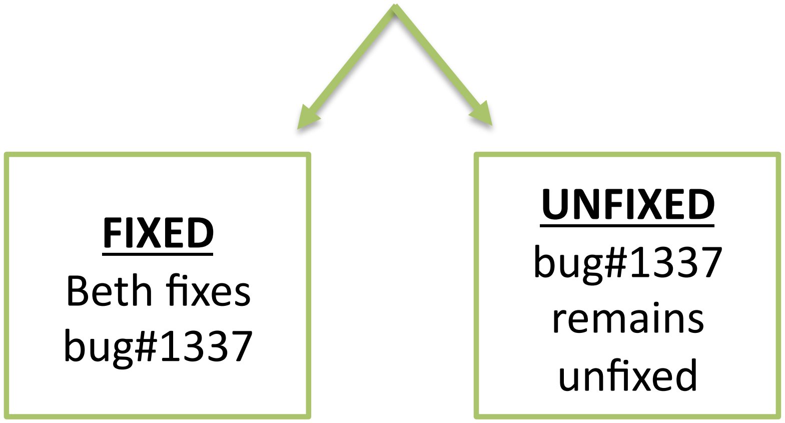
Beth

Courtesy Georg Link

Step 4: Two weeks pass

- Adam is waiting
- Beth is developing

- Two possible outcomes



FIXED
Beth fixes bug#1337

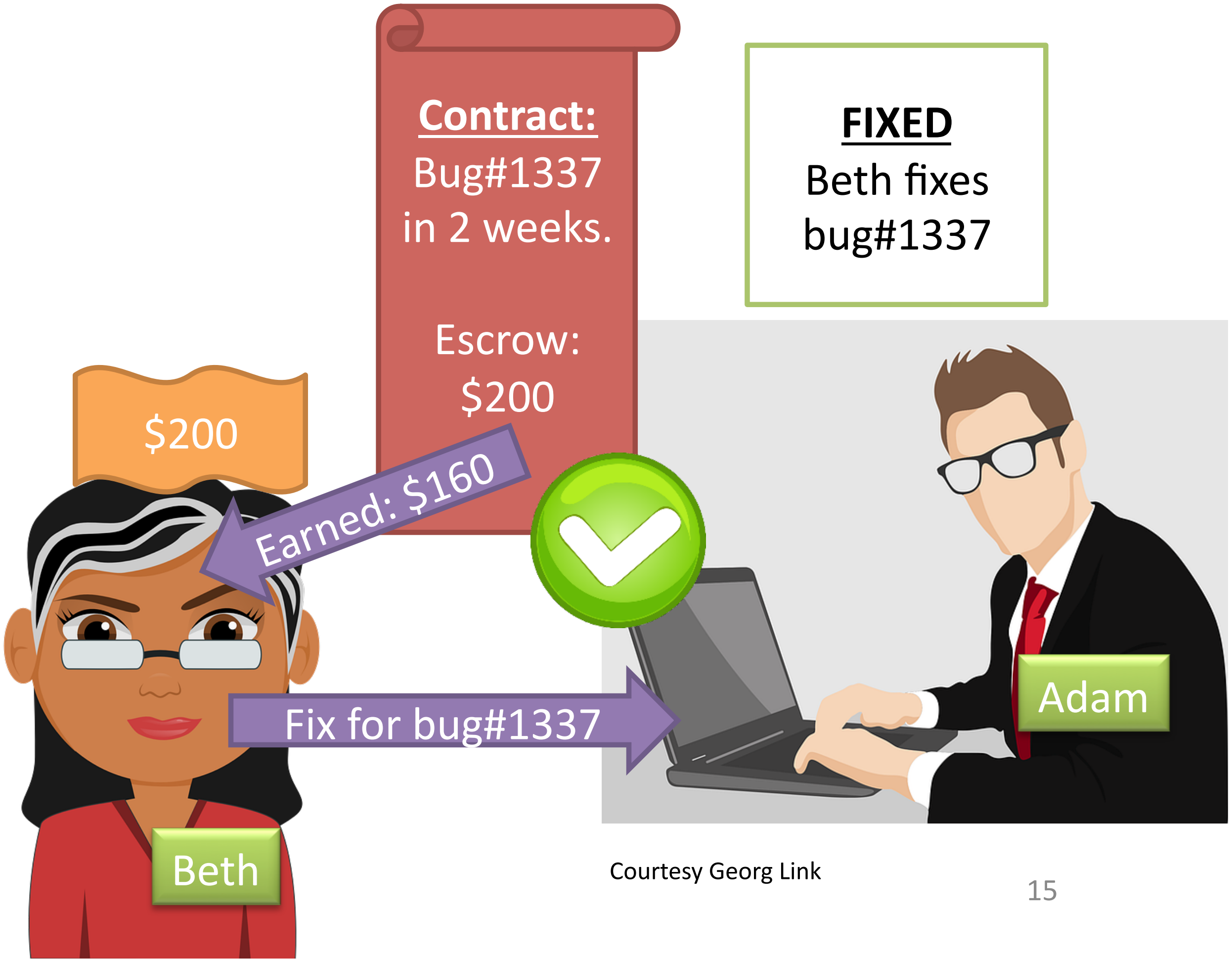
UNFIXED
bug#1337 remains unfixed

Courtesy Georg Link

Outcome: FIXED

- The issue is closed
- Beth earns the reward of $160 and gets her $40 deposit back
- Adam receives the fix and pays with his deposit

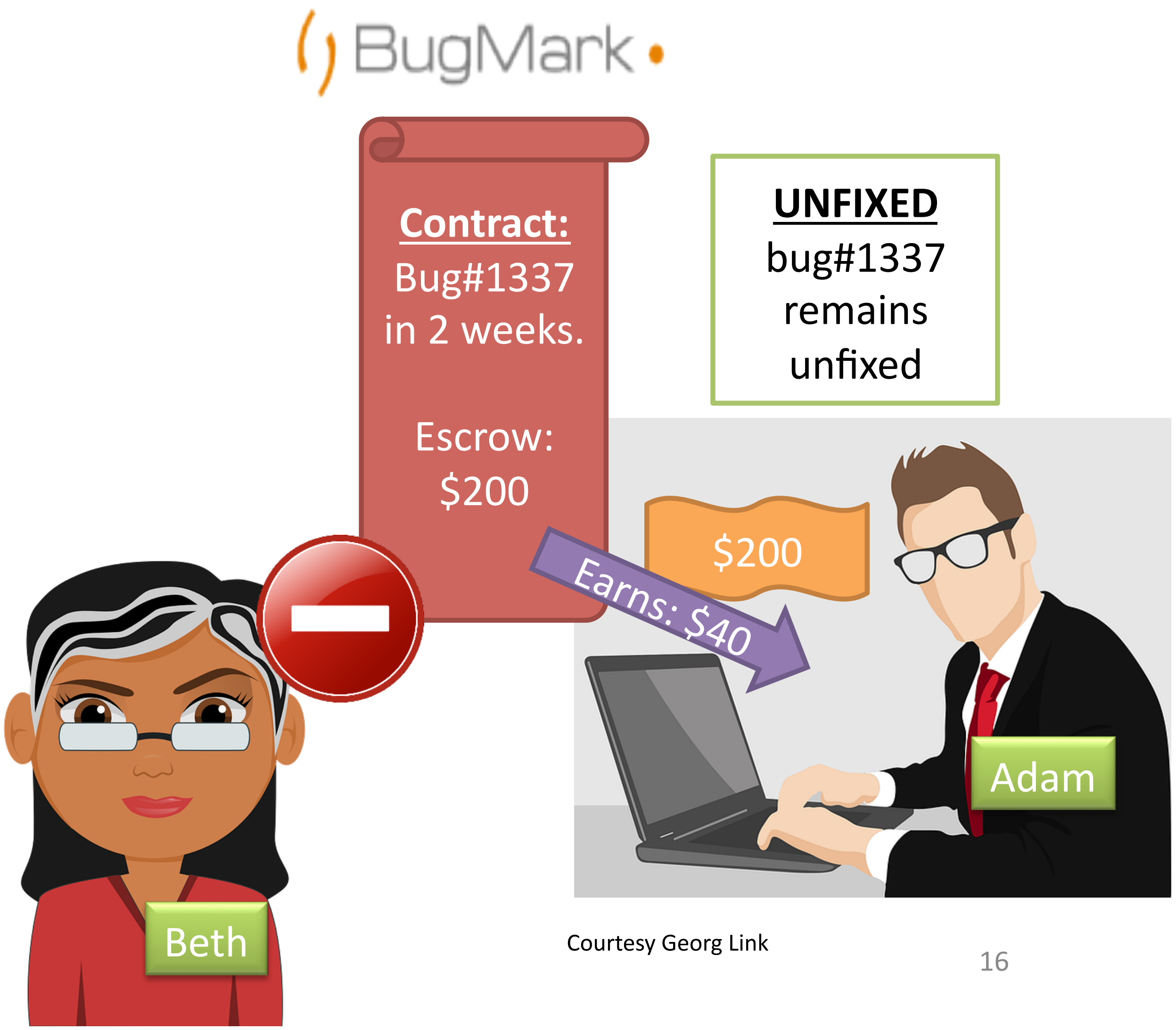
BugMark



Courtesy Georg Link

Outcome: UNFIXED

- The issue is open
- Beth loses her $40 deposit
- Adam receives his and Beth's deposits: earns $40



Contract:
Bug#1337
in 2 weeks.

Escrow:
$200

UNFIXED
bug#1337
remains
unfixed

Courtesy Georg Link

Design features

- Partial work
- Dependencies
- Indemnity
- Decoupling funding from work
- Multiple contracts

Our approach

- How are we different from existing systems?
  - Bounty systems: typically reward the submitter of the final solution. But development is often done in a collaborative way where work layers on previous work [Howison and Crowston 2014]. Tasks may also require different skills. Need a way to assign credit to various contributors involved in a solution.

    Trading market allows users to do partial work on a contract and then resell their positions.

Our approach

- How are we different from existing systems?
  - Prediction markets: small number of tradable questions but a large number of participants ("wisdom of the crowd").

    We have a large number of futures contracts but a small number of participants per tradable task ("wisdom of individuals" revealed to crowds of software users).

    Participants have information about the issue and can influence the outcome.

Outline

- Introduction
- Background and motivation
- Our approach
- Simulation
- Ongoing and future work
- Conclusion

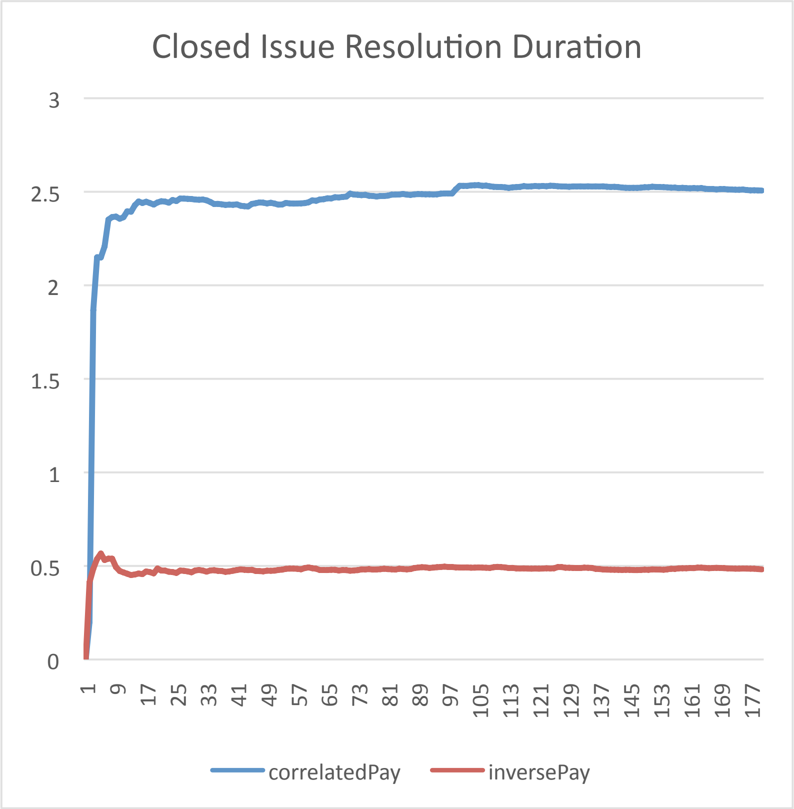
Simulation

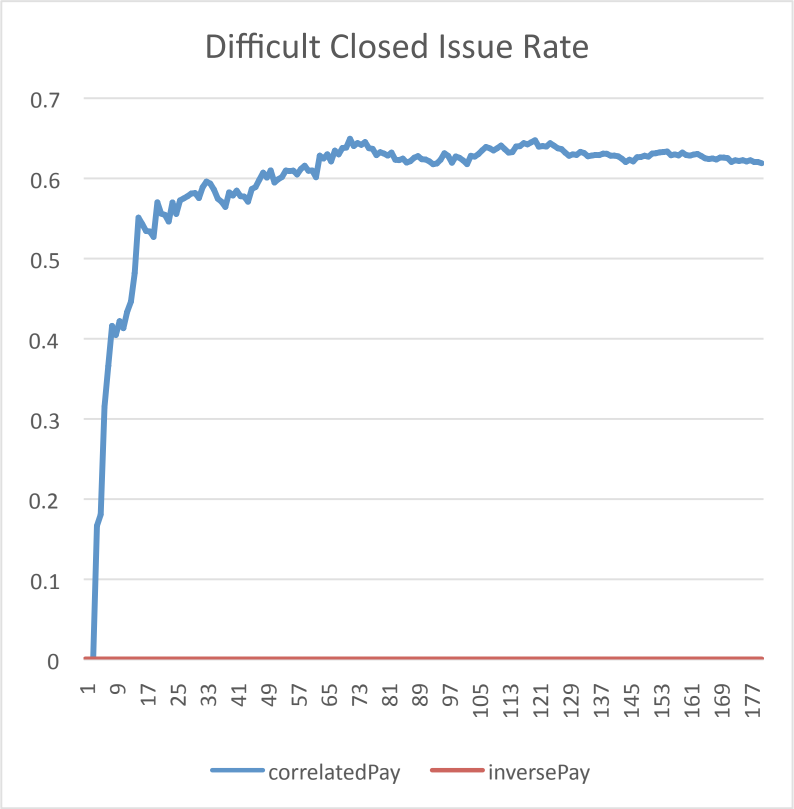
- **2** funders each owning a project: **correlatedPay and inversePay**
- **10 workers** use price signals to choose issues.
- Each day, **each funder generates 10 new issues** and an accompanying offer that **matures after three days**.
- Issues are randomly assigned a **difficulty level of 1, 2, or 3** (1/3 probability each) which indicates how many days it takes for a worker to finish the issue. Every issue has **40% chance to require 1 extra day** to finish.

Simulation

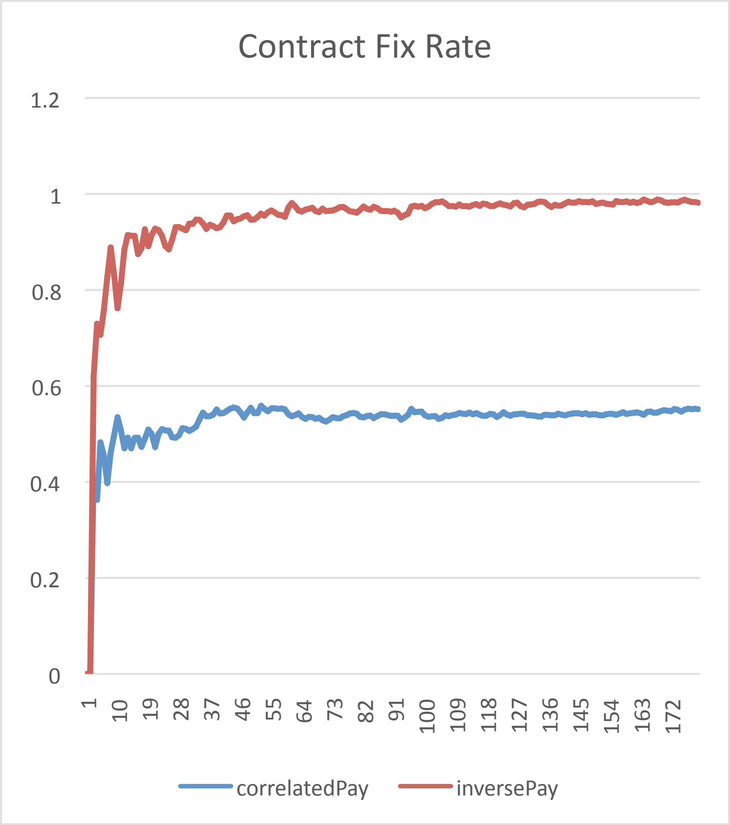
- **Unit prices are 0.9, 0.8, and 0.7 tokens.** CorrelatedPay offers higher prices for more difficult issues (i.e., 1 -> 0.7, 2 -> 0.8, 3 -> 0.9). InversePay offers higher prices for easier issues (i.e., 1 -> 0.9, 2 -> 0.8, 3 -> 0.7).
- Every offer is for the **volume of 100 units and pays out 100 tokens**. The price for a buy unfixed offer of 0.9 means that the funder pays 90 tokens and the worker 10 tokens.
- The simulation **runs for 180 days and is repeated 5 times**. Values presented are averages over all 5 runs.

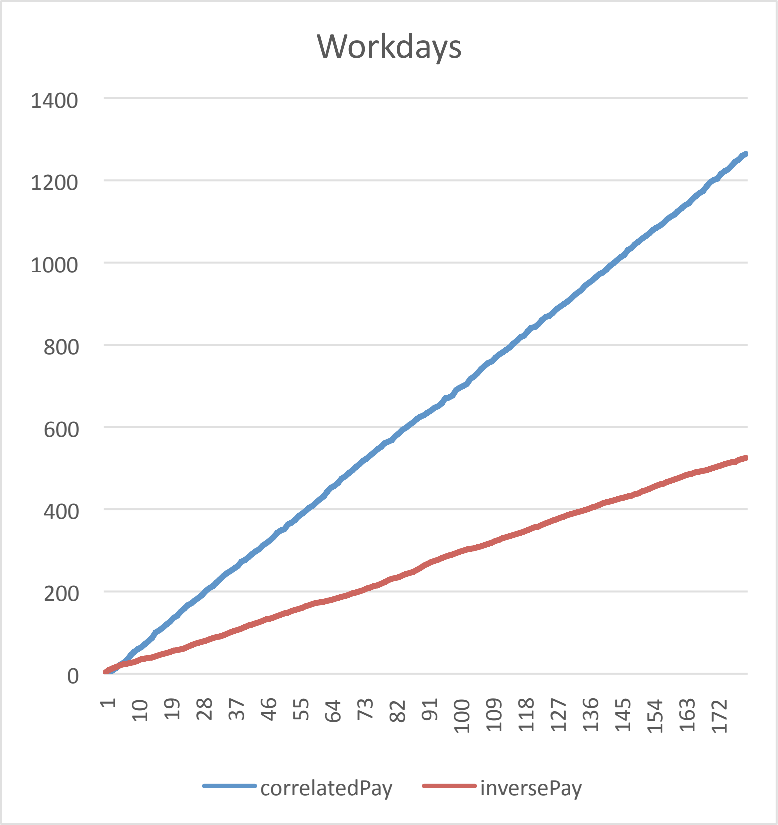
Simulation results



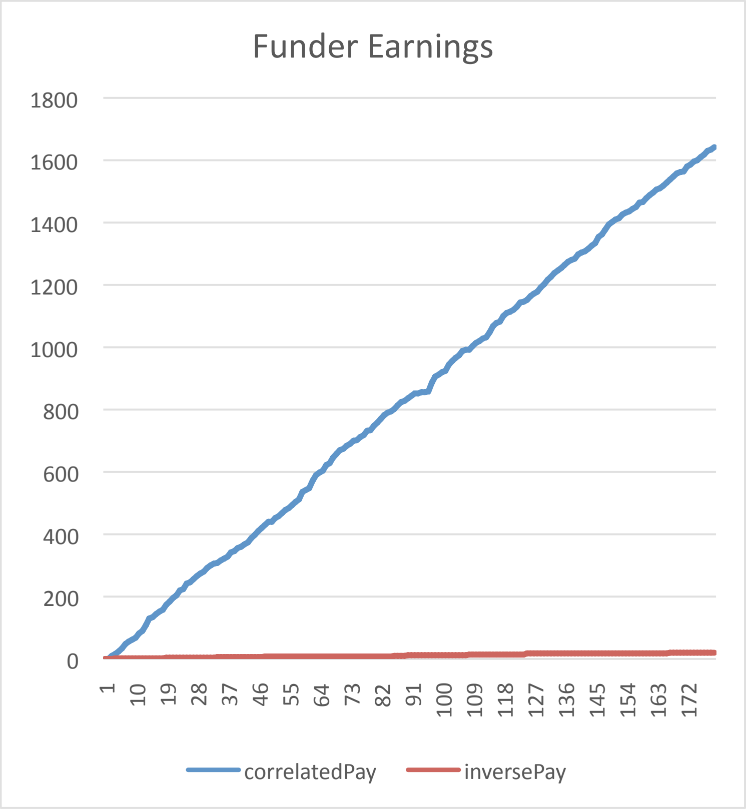


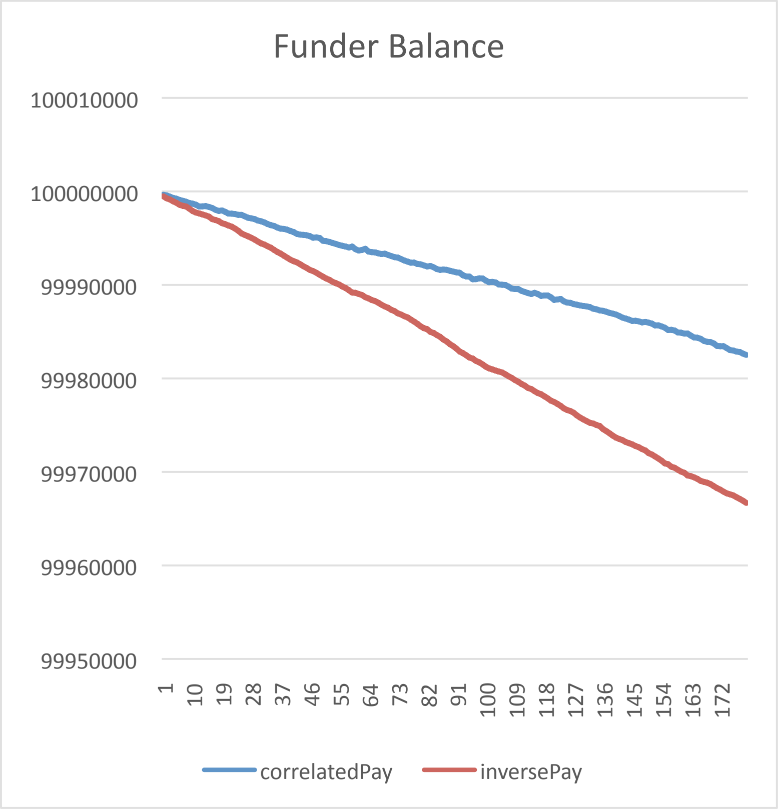
Simulation results





Simulation results





Outline

- Introduction
- Background and motivation
- Our approach
- Simulation
- Ongoing and future work
- Conclusion

Ongoing and future work

- What are the consequences of introducing price signals in peer production?
- How does this model compare to other market models?
- Scope of futures markets and their application to the contract design.
- Interaction of market with current secure software engineering practices

Ongoing and future work

- How can such a system work alongside other diverse motivations to contribute?
- How to incentivize project maintainers and reviewers, incorporate testing, verify work is done correctly?
- Lots of experiments and data needed
  - Completed live worker-funder-trader usability study
  - More simulations and studies to follow
  - Collaborations welcome

Conclusion

- A futures trading market to incentivize software tasks in peer production communities
- Market design that introduces price signals in a way that leverages some of the successful qualities of peer production
- Credit for partial work enables information sharing and collaboration
- The market treats a prediction that something cannot be done in the same way as an incentive to make it happen

Thank you

malvikar@gmail.com