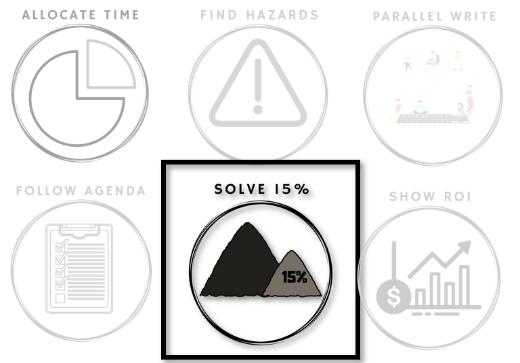
Solve 15%



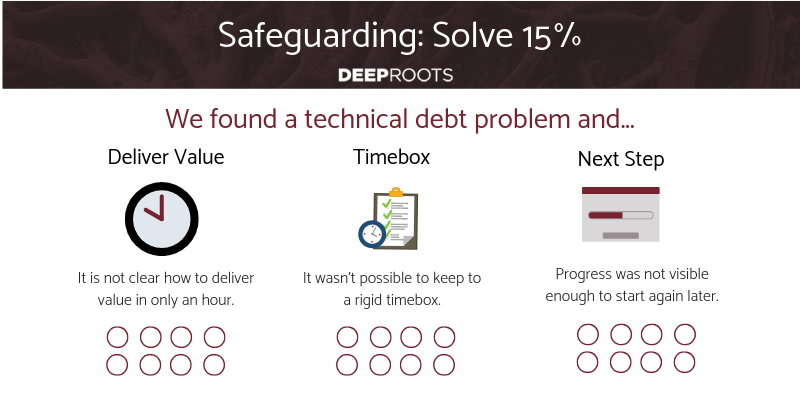
## How can we get partial return from partial investment?

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | RCAs are all about carefully analyzing your Hazards. Fully addressing those Hazards is expensive. Your RoI depends on the quality of your analysis.  Safeguarding is different, and that difference arises because we Solve 15%.  Safeguarding maximizes RoI by spreading your investment. Each bug funds a small investment to partially address the Hazards that led to it—to make them 15% better. When bugs arise from a common Hazard, each bug will fund a partial fix for that Hazard. Over time, the investments cluster on whichever Hazards result in the most problems.  You don't need to analyze your Hazards and guess which is most significant. Your stream of bugs automatically guides you to invest more where you will have the best return.  Getting this amortized return requires that you can get a small return from each small investment. It also requires that you can set down an investment for a while, then pick it up to make the next investment weeks later. Your team needs to be able to work incrementally on technical debt. Many teams struggle with this.  There are many ways a team can fail to solve technical debt incrementally. This week is about finding your team's pattern. | | | |

Beginning of Week: Experiment

This week look at what happens after we find a technical debt problem.

Each time you find a technical debt problem, look for the three symptoms on the tracking image below and mark which one(s) appeared.  
  
At the end of the week, you will look for patterns in your team's symptoms.



End of Week: Reflect

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | Getting value from small investments requires discipline. Well-running teams can set a timebox as short at 15 minutes and always check in value, even on complex redesigns. And they can pick up efforts from months ago without duplicating effort. How does your iterative ability compare?  Which symptoms impacted you the most often?  Do you struggle to check in value each hour? Do you know how to make progress visible? If you were to pick a random moment in the middle of one of your changes, what would prevent you from abandoning all work since the last commit?  Can you solve your most significant symptoms yourself?  Our 15% solution techniques solve all 3 symptoms at once. Do you want help? | | | |



Daily Stand-Up Questions

1. What work do you have in progress?
2. What blocked you from checking in before this meeting?
3. How could you change that?

   
Weekly Retrospective Questions

*This is intended to take a full 60 minute retrospective meeting.*

1. What is a problem that remains after many retrospectives?
2. For that problem, how could you make partial progress visible?
3. How could you work for 1 hour on this problem, walk away, and make 1 hour of progress?