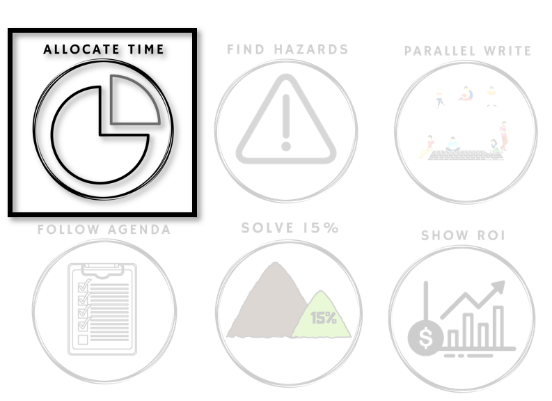
Allocate Time



## How do we agree how to spend time on

## technical waste — in advance?

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | Technical debt will usually be found by an individual developer while working alone. At that moment, that individual will make a choice - often subconsciously - how to allocate the next chunk of time between the newly-found debt and the prior task.  If your whole team - developers, product owners, and managers - are aligned in advance, then those decisions will match your strategy. If not, then those decisions are likely based on personal bias. Aligned decisions will accumulate where individual decisions will cancel each other out.  There are many ways a team can fail to allocate time in an aligned way. This week is about finding your team's pattern. | | | |

Beginning of Week: Experiment

Throughout the week you will see technical waste appear. Examples will include:

* Long method
* Hard-to-test class
* Non-obvious code

Each time you see technical waste, 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

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | Most developers run into 10-20 technical debts each day, but have learned to not notice most of them. How many were you able to see?  Which symptoms impacted you the most often?  As a result, how often did you choose to fix the debt?  How often this week did you find yourself in a trap between choosing to do what you thought was right for the code vs doing what you thought your Product Owner or manager would want?  Can you solve your most significant symptoms yourself?  Safeguarding budgeting addresses all 3 of these symptoms. Do you want help? | | | |



Daily Stand-Up Questions

1. How many times in the last day did you notice tech debt?
2. Share why you chose to resolve it or not.

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

1. Sticky list tech debt from the week.
2. Mark which ones will be / were fixed.
3. How will management reward each possible fix? 
4. Find the pattern of what is and isn’t rewarded.