# Myth #1: Pipeline Improves Quality



# Developer Roadmap

## Strategy #1: Safeguarding

[**bit.ly/safeguarding-steps**](bit.ly/safeguarding-steps)

|  |
| --- |
| How would you implement safeguarding? |
|  |
| What impediments do you need to solve? |
|  |
| What action is needed to remove impediments? |
|  |

## Strategy #2: Refactor Scary Code

|  |
| --- |
| What has blocked you from refactoring this code already? |
|  |
| How can you address those blocks? |
|  |
| What small changes can you do to make it less scary? |
|  |

## Strategy #3: Increase Discipline

|  |
| --- |
| Watch Arlo Belshee’s DeConstruct Talk<https://www.deconstructconf.com/2017/arlo-belshee-i-find-bugs-too-boring-to-write>  |
| What tools do you have for disciplined refactoring? |
|  |
| How could you make your refactoring safer? |
|  |

# Leader Roadmap

## Strategy #1: Empower Devs

|  |
| --- |
| What do you believe is stopping your devs from refactoring?  |
|  |
| How would you validate or invalidate your assumption?  |
|  |
| Assuming you’re right, how would you fix it?  |
|  |

## Strategy #2: Safety Culture

|  |
| --- |
| Watch ALCOA’s Safety VideoReplace aluminum with code.Replace injury with bug.<https://nyti.ms/2jRMI7s>  |
| How could you establish this habit at your company? |
|  |
| What actions or systems are you perpetuating that might block safety? |
|  |
| What are the indicators that will demonstrate significant culture change around bugs? |
|  |

# Myth #2: Pipeline Helps Me Recover Quickly



# Developer Roadmap

## Strategy #1: Acceptable Bug Definition

|  |
| --- |
| What indicators will tell you it is time to narrow the set of bugs you “won’t fix”? |
|  |
|  How could you notice the unseen costs of the bugs you are currently not fixing? |
|  |
| What processes should you change or remove when narrowing the “won’t fix” bugs definition?  |
|  |

# Leader Roadmap

## Strategy #1: Safety Definition

|  |
| --- |
| Assuming safety culture for bugs from Myth 1 is established, what are other risks to consider for safety?  |
|  |
| How can you leverage the safety culture you’ve built for bugs to address these additional risks?  |
|  |

# Myth #3: Invest in Tools



# Developer Roadmap

## Strategy #1: Refactoring Mentor

|  |
| --- |
| Who do you know who has refactored untested code well?  |
|  |
| If you don’t know anybody, inquire in the Legacy Code Rocks slack channel. <https://www.legacycode.rocks/community>  |
|  |
| What problems do you need to know how to solve?  |
|  |

# Leader Roadmap

## Strategy #1: Technical Skills Roadmap

|  |
| --- |
| Which experienced developers can tell you the sequence of skills you need on your roadmap?  |
|  |
| Do you have the people necessary to make a center of technical excellence? |
|  |
| What gaps do you need to fill with hiring or consultants?  |
|  |

# Myth #4: Prioritize Adding Unit Tests



# Developer Roadmap

## Strategy #1: Code Issue Visibility

|  |
| --- |
| How do you currently hide technical waste from managers and others?  |
|  |
| What is the source of the unsafety that causes you to hide it?  |
|  |
| What would you need to feel safe?  |
|  |
| If you felt safe, what would you do to make the code problems visible to the entire organization? |
|  |

## Strategy #2: Prioritization

|  |
| --- |
| What do you need the manager to do that shows that the code problems are a real priority?  |
|  |
| If the code problems are a real priority, what would be your actions?  |
|  |
| What benefits do you expect those actions to give the company in what timeframe?  |
|  |

##

# Leader Roadmap

## Strategy #1: Alignment

|  |
| --- |
| Who are the right leaders for the “fix the ugly code” project?  |
|  |
| What structural support would they need to make that project succeed?  |
|  |
| How do you change annual review to make this priority clear?  |
|  |

# Prioritization Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Priority | Myth | Strategy | Action | Timeframe |
| [choose level 1, 2, or 3] | [choose 1, 2, 3, or 4] | [choose 1, 2, or 3]  | [write your specific action] | [write your personal due date] |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |