“But EVERYTHING is High Priority!”

How to prioritize Epics and Features with External Clients
How Developers Prioritize

- Cool Stuff
  - Prototypes
  - Innovation
  - New Technologies

- Code Quality
  - Architectural Runway and Design
  - Bug Work
  - Tech Debt
  - Performance

- New Capabilities
  - Epics
  - Features
  - User Requests

- Boring but Necessary
  - Unit Tests
  - Documentation
  - Knowledge Transfer
How Clients Prioritize

- New Capability
  - Epics
  - Features
  - User Requests

- Usability
  - Bug Work
  - Performance Improvements
  - Documentation

- Expensive but unnoticeable
  - Architectural Runway and Design
  - Prototypes and Innovation
  - Unit Tests
  - Knowledge Transfer
So how do you meet client expectations while satisfying developer needs?
So how do you meet client expectations while satisfying developer needs?

Set consistent boundaries ahead of time

- New Capabilities: 60%
- Code Quality: 15%
- Innovation and Planning: 15%
- Devs Choice: 10%
So how do you meet client expectations while satisfying developer needs?

- Set consistent boundaries ahead of time
- Make a priority list, not priority groups

New Capabilities: 60%
- Code Quality: 15%
- Innovation and Planning: 15%
- Devs Choice: 10%
So how do you meet client expectations while satisfying developer needs?

Set consistent boundaries ahead of time

Make a priority list, not priority groups

Prioritize from largest to smallest containers of work

<table>
<thead>
<tr>
<th>New Capabilities</th>
<th>Innovation and Planning</th>
<th>Code Quality</th>
<th>Devs Choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>15%</td>
<td>15%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Epics
1. Epic 1
2. Epic 2
3. Epic 3
4. Epic 4

Features
1. Feature A
2. Feature B
3. Feature C
4. Feature D

Epics and Features
1. Epic 1
   1a. Feature A
   1b. Feature B
2. Epic 2
   2a. Feature C
   2b. Feature D
   2c. Feature E
3. Epic 3
   3a. Feature F