

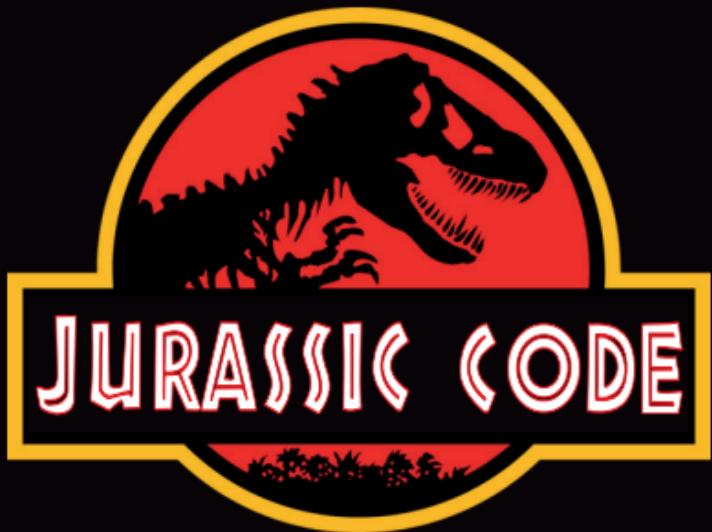


GATHER PRODUCT INSIGHTS

Identify which features are covered by this product.



Product Backlog
User Stories
Acceptance Criteria



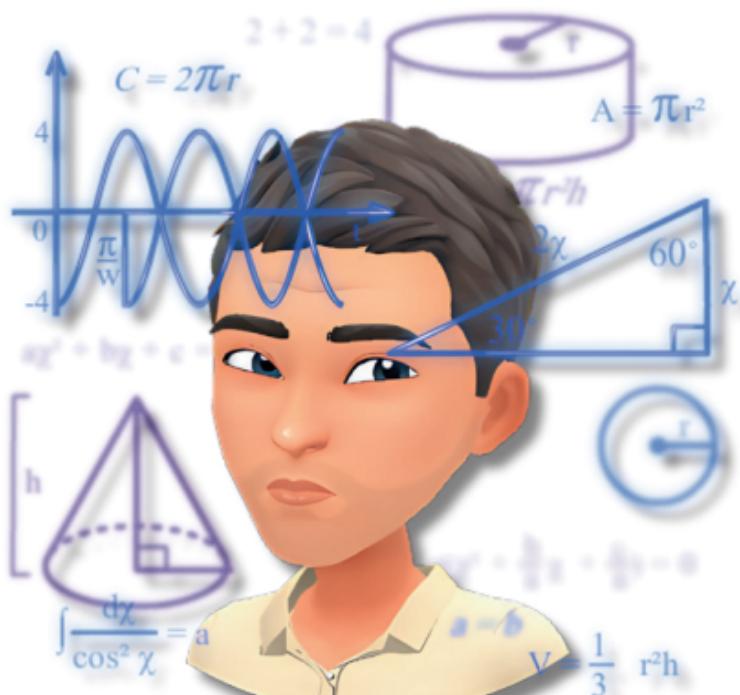
<https://bit.ly/42XQZNh>



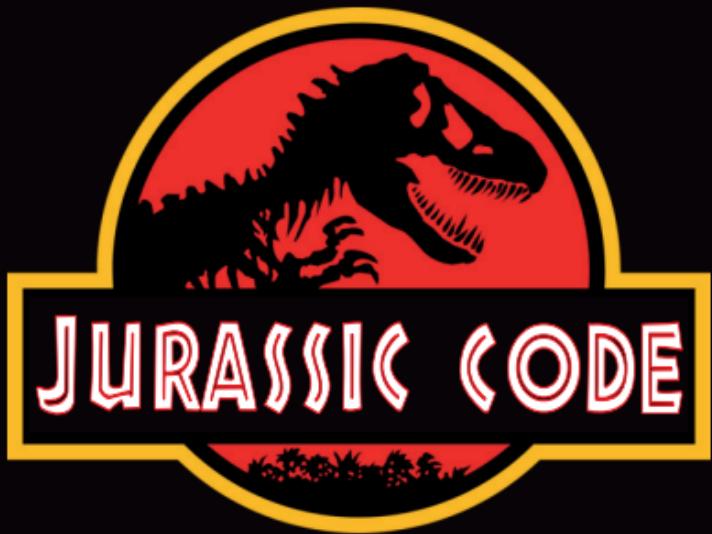
TAMING THE DINOSAURS
OF LEGACY CODE

EXPLAIN THE ARCHITECTURE

Deep dive into the Software Architecture.



C4 Model
Mermaid diagrams



<https://bit.ly/3Shwc2j>

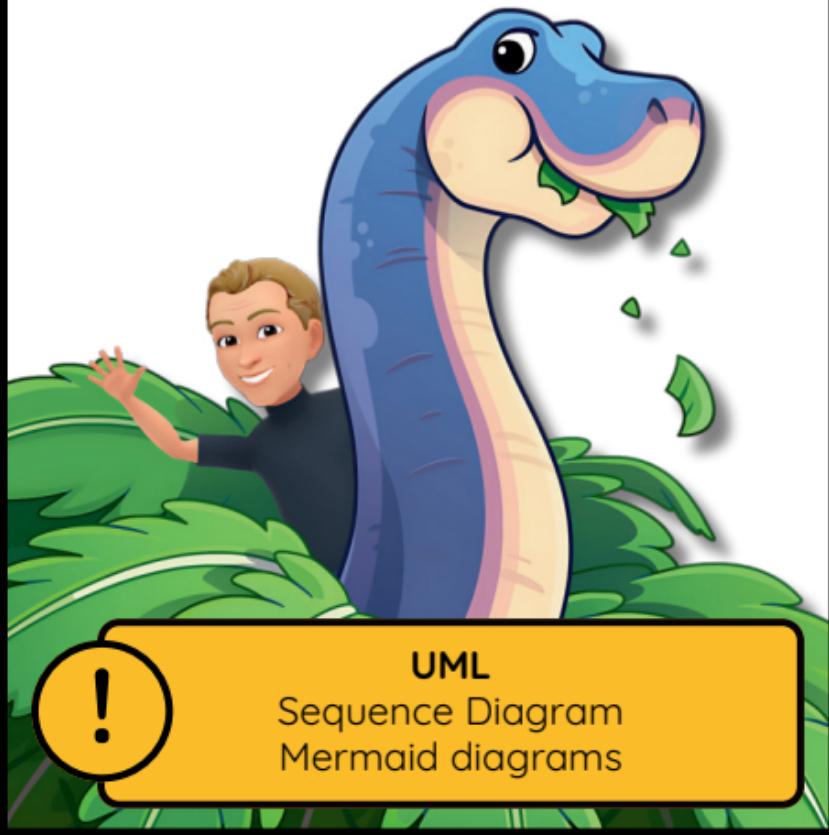


TAMING THE DINOSAURS
OF LEGACY CODE

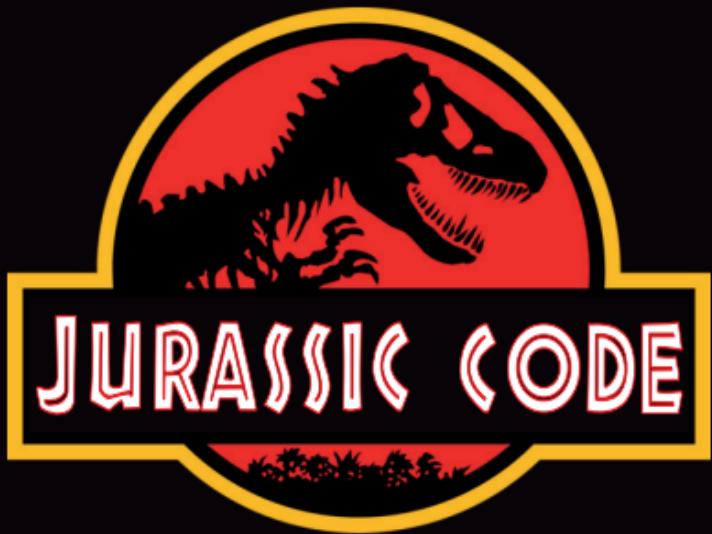


DETAIL A FLOW

Explain in-depth a feature implementation from front to back.



UML
Sequence Diagram
Mermaid diagrams



<https://bit.ly/3YZ4N9a>



TAMING THE DINOSAURS
OF LEGACY CODE



RATE CODE QUALITY

Score code quality.



Code smells
Clean Code



<https://bit.ly/3GCrSs9>



TAMING THE DINOSAURS
OF LEGACY CODE

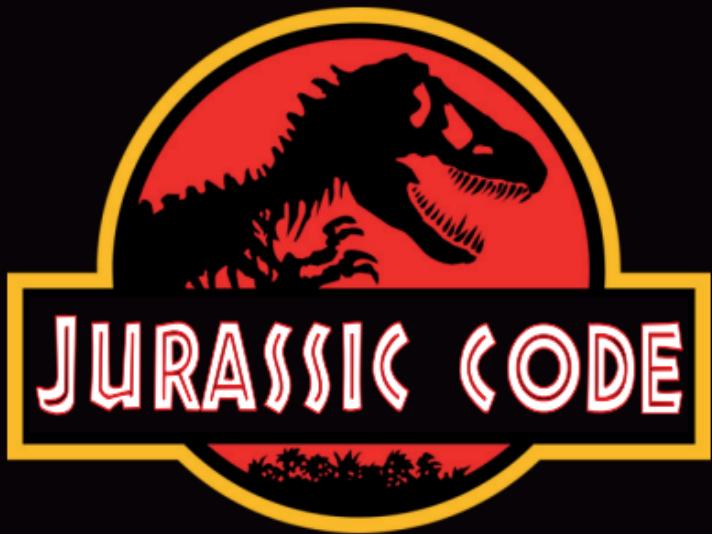


GENERATE SOME TESTS

Increase code coverage
before iterating on this code.



3A pattern
Test Data Builder
Fluent business assertions



<https://bit.ly/4jDihjf>

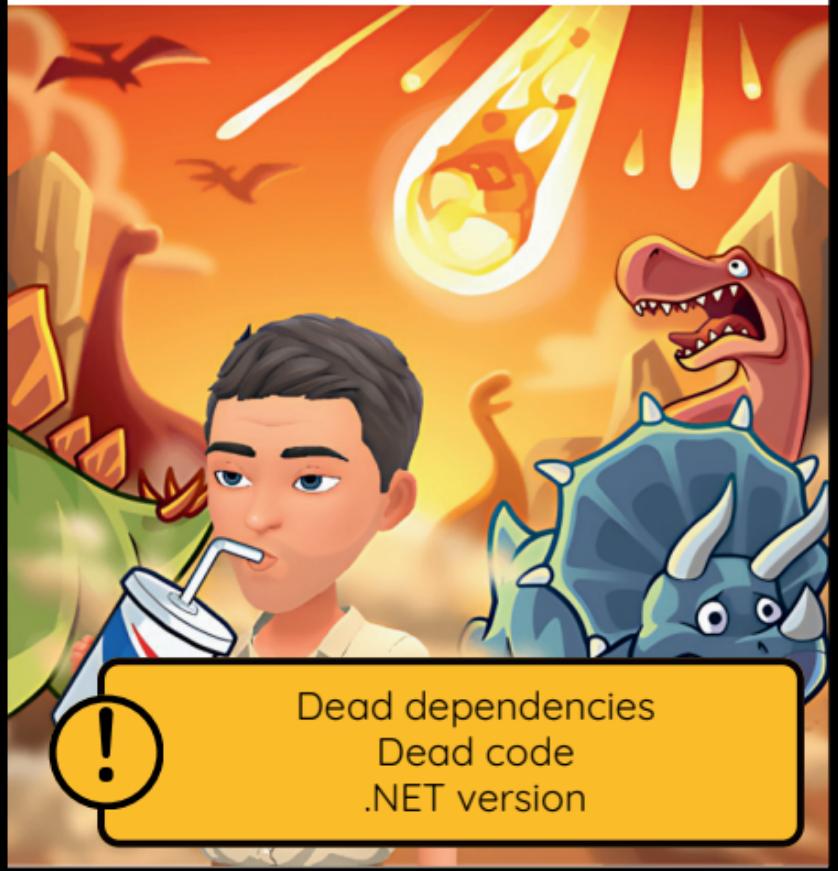


TAMING THE DINOSAURS
OF LEGACY CODE

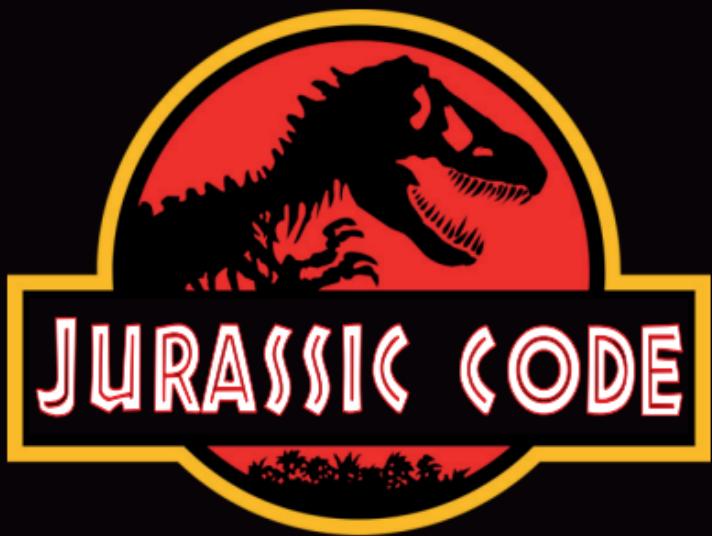


MAKE SOME CLEAN UP

Make a bit of cleanup in the code.



Dead dependencies
Dead code
.NET version



<https://bit.ly/4jwbAzs>



TAMING THE DINOSAURS
OF LEGACY CODE

USE CLEAN ARCHITECTURE

Refactor the code to use
Clean Architecture principles.



Dependency inversion
Use Cases
Screaming Architecture



<https://bit.ly/4d0DMrV>



TAMING THE DINOSAURS
OF LEGACY CODE

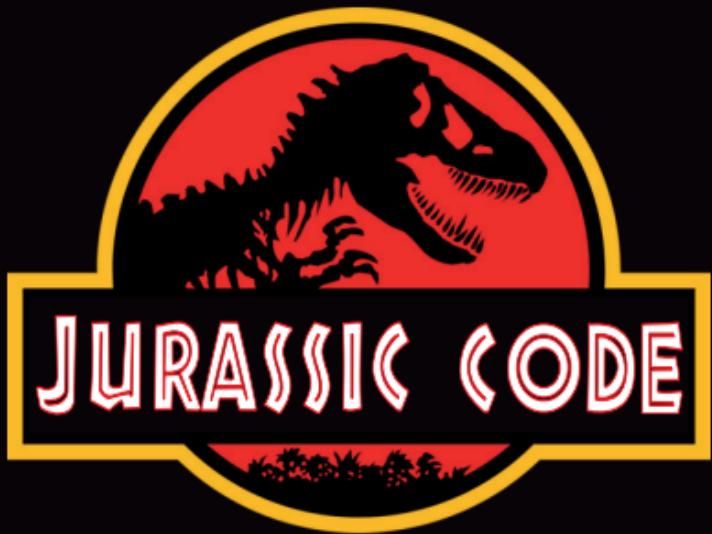


REFACTOR SOME CODE

Understand quickly what / how to refactor a given piece of code.



Mikado method
Sprout Method



<https://bit.ly/4d22Hvc>



TAMING THE DINOSAURS
OF LEGACY CODE