Legacy to Legendary

Sustainable patterns and practices for app modernisation

Thank You to Our Sponsors and Hosts!

OWASP NEW ZEALAND owasp.org.nz

Waipapa Taumata Rau University of Auckland

AppSec NZ appsec.org.nz

BASTION SECURITY GROUP

SECURE CODE WARRIOR®

exabeam™

safe advisory.

CyberCX

plexure

SECDIM

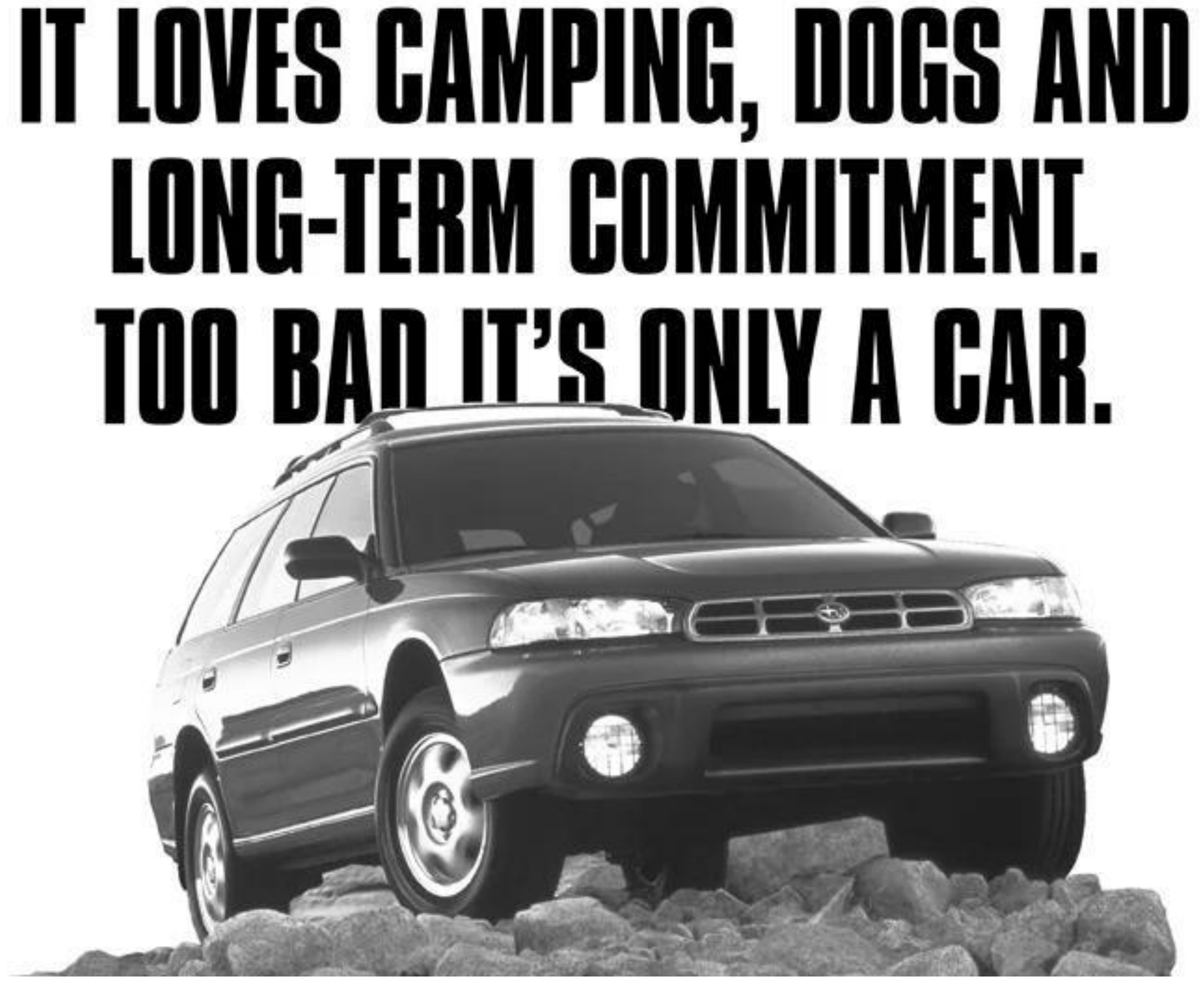
Without them, this conference couldn't happen.

- Some definitions
- Why it’s important to know how to work with legacy code
- A couple of useful patterns for working with legacy
- Practices for working with legacy, including the people and teams
- The connection between working with legacy code and AI-assisted coding
- Conclusions and closing thoughts

Overview and themes

https://www.npr.org/2022/06/29/1108739853/when-subaru-came-out-classic

What is legacy anyway?



What’s of paramount importance is working software, especially working software that people use and ideally love (Agile Manifesto)

My definition:
Any code that is in production

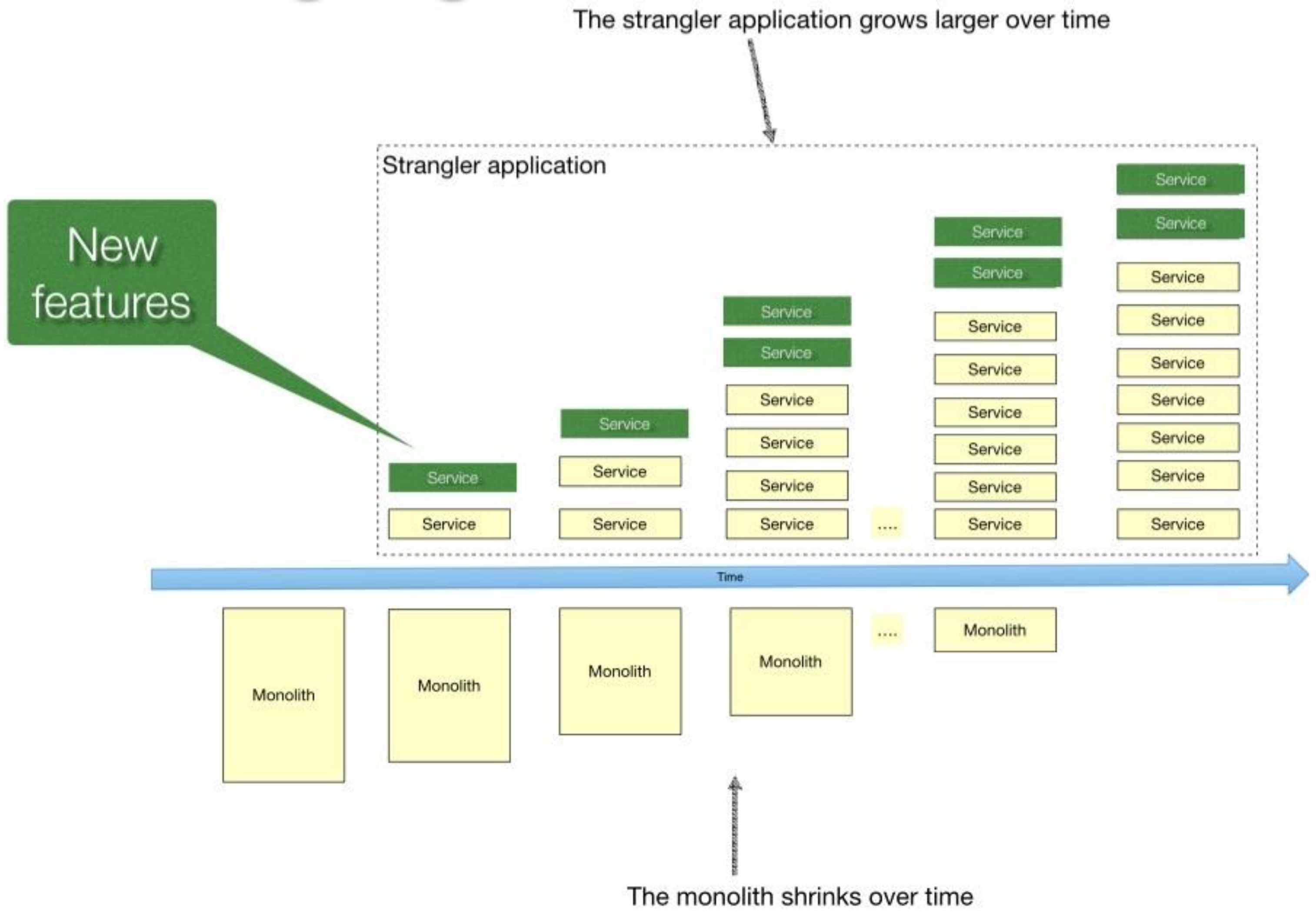
Michael Feathers’ definition:
Any code that is not tested

What is legacy anyway?

Why I love legacy

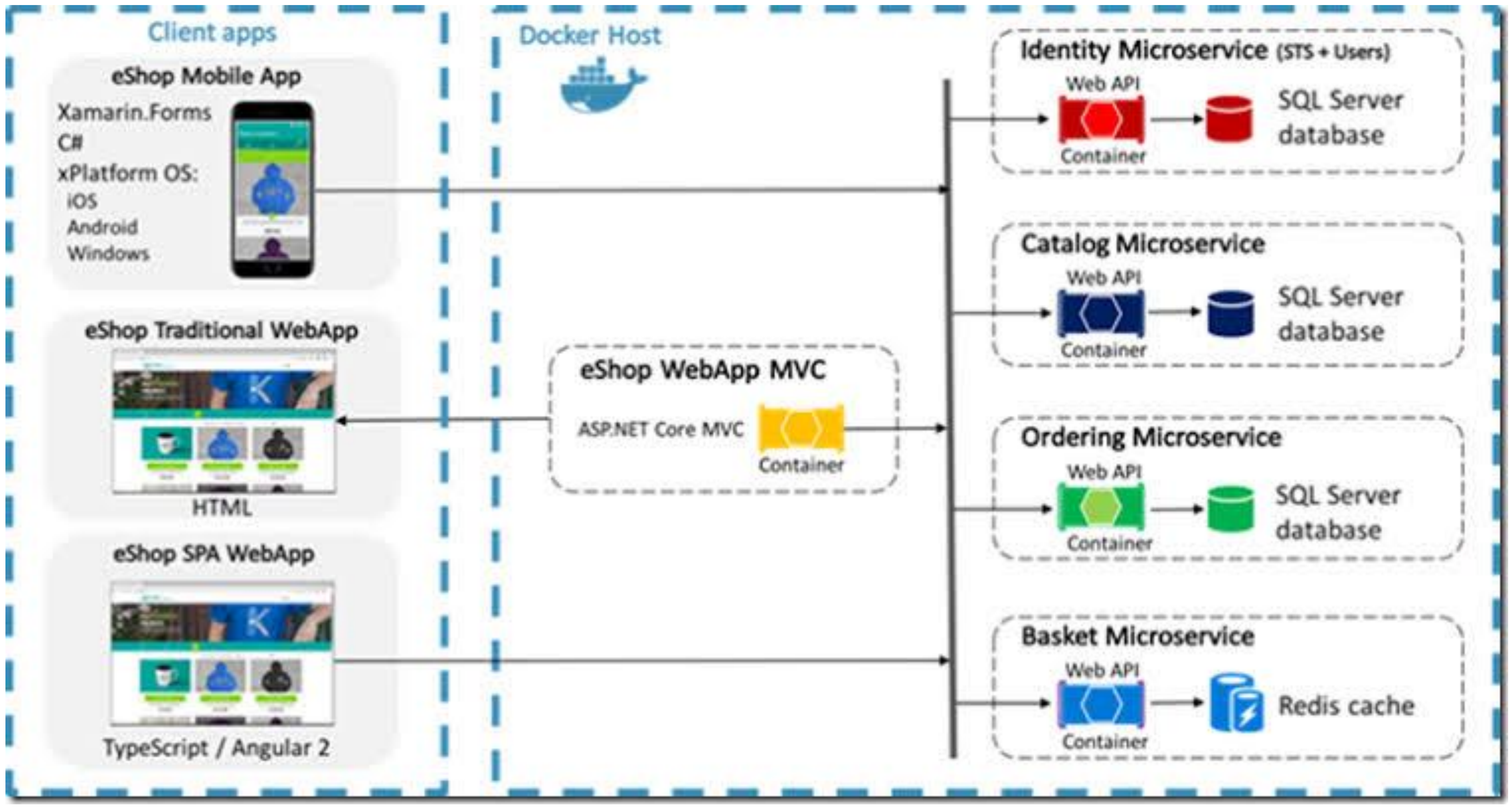
https://www.goodreads.com/book/show/2718384-more-joel-on-software

Patterns: Strangler Fig Application (Martin Fowler)


The strangler application grows larger over time
Strangler application
New features
Service
Time
....
Monolith
The monolith shrinks over time

Patterns: Strangler Fig Application (Martin Fowler)

https://microservices.io/patterns/refactoring/strangler-application.html

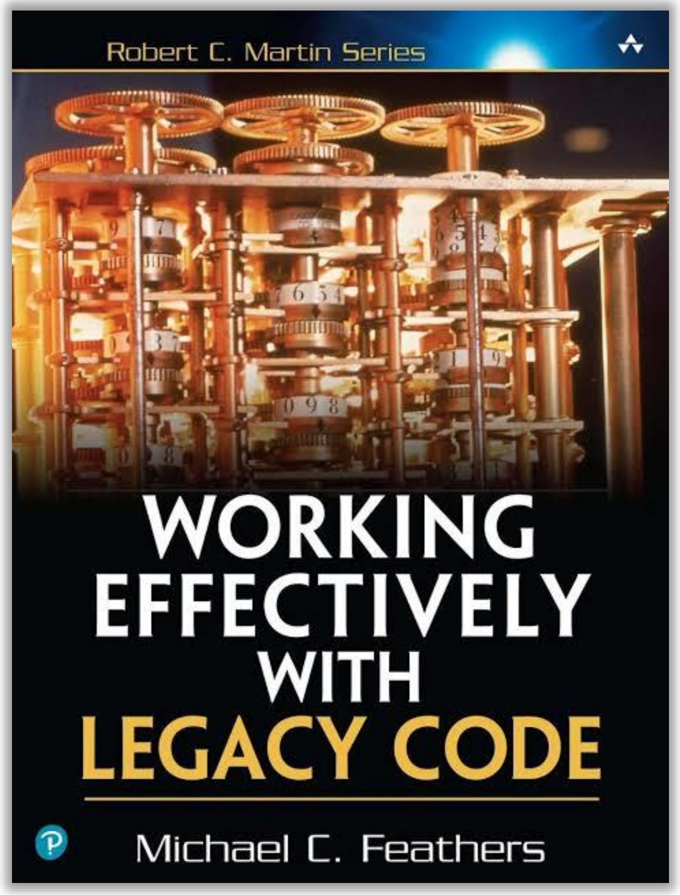


https://devblogs.microsoft.com/cesardelatorre/free-ebookguide-on-net-microservices-architecture-for-containerized-net-applications/

Patterns: Seams (Michael Feathers)

https://www.nzgeo.com/stories/the-obsidian-island/

1. Identify seams in the codebase, ideally where domains converge (bounded context pattern)
2. Break dependencies across seams by introducing tests (dependency injection pattern)
3. Change legacy code safely and incrementally, automating test and release as-you-go
4. Manage/reduce fear in the team
5. Repeat…



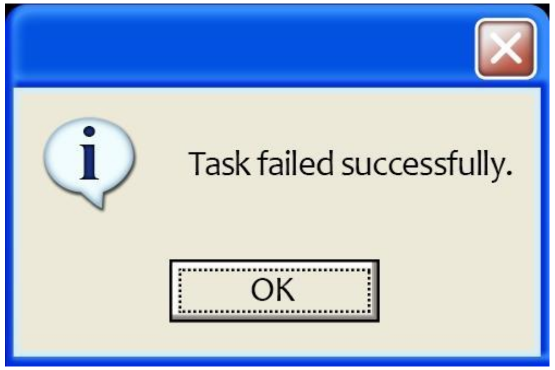
The playbook for refactoring legacy code according to Michael Feathers

https://www.goodreads.com/book/show/44919.Working_Effectively_with_Legacy_Code

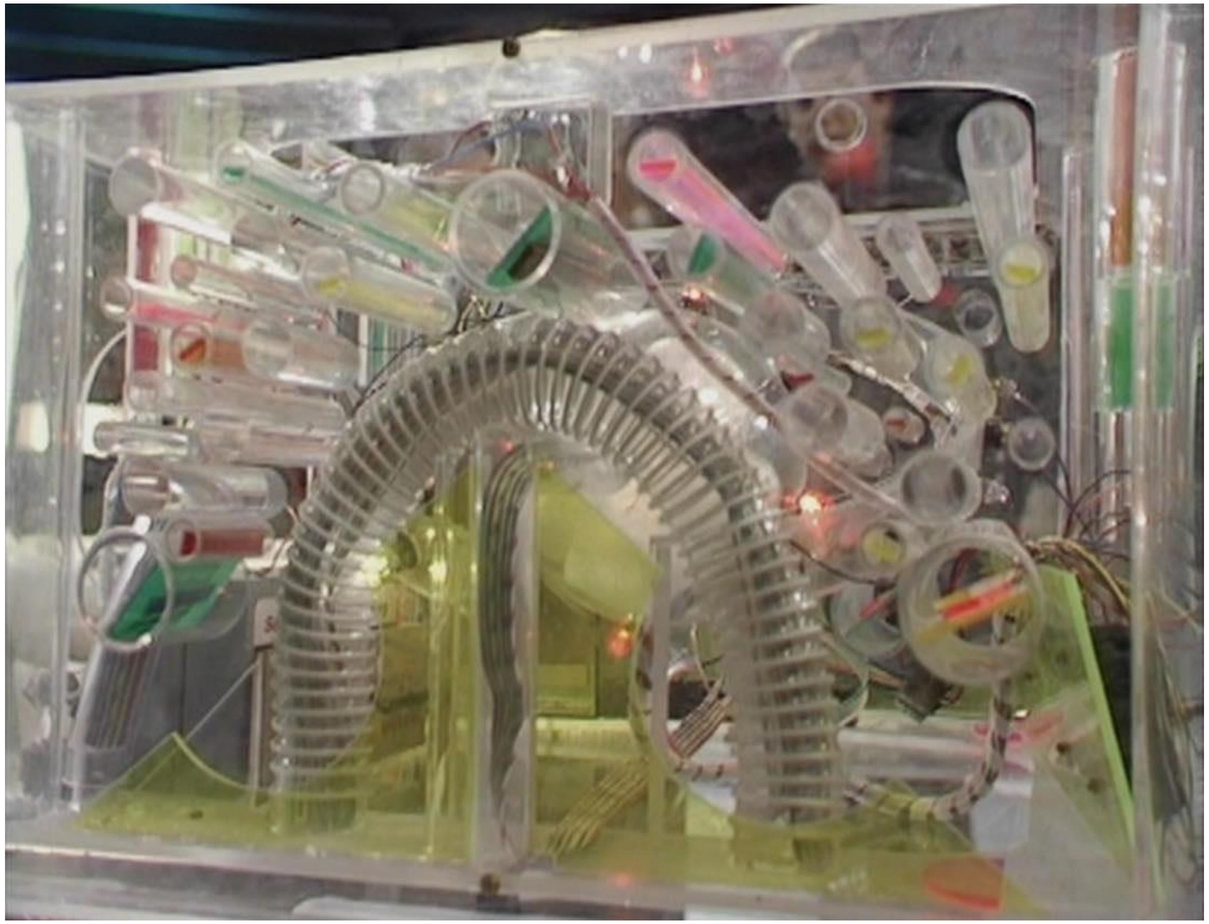
Practices: table stakes stuff

https://knowyourmeme.com/memes/this-is-fine

- Help your team to see and play the long game – anyone can do this (doesn’t need to be the boss)
- Put a modernisation roadmap together and have clear, simple milestones, that target customer-led initiatives (and maybe a little zhuzh)
- Experiment with and use the emergent AI tools that specifically target legacy modernisation use-cases


Task failed successfully.

Practices: pragmatism, excitement, innovation



A perfect match: gen-AI & legacy modernisation

https://blakes7.fandom.com/wiki/Orac

- Don’t be scared of legacy - learn to love it; it’s healthy
- Use well established patterns and practices to get a grip on your legacy code
- Things have changed since Joel Spolsky’s blog post, but the underlying principals still hold – try not to reinvent the wheel
- There are strong parallels between AI-assisted coding and dealing with legacy
- We’re probably just at the beginning of how AI will affect and benefit legacy modernisation

Conclusions