# A developer always pays their tech debts

**Wesley Lomax & Jonathan Robbins**

April 24, 2018

#sugcon

# Who put all this Technical Debt here?!

# Should you pay your debts?

# Can I throw everything away and start over?

Not all hope is lost!

Take the IA with you?

Webforms? More like Lameforms. Kill the presentation only

Breaking point between trying to recover and starting again

# Well why should I make work for myself?

Raise the quality limit

Code decay

Short term pains for long term gains

Preempt serious rework

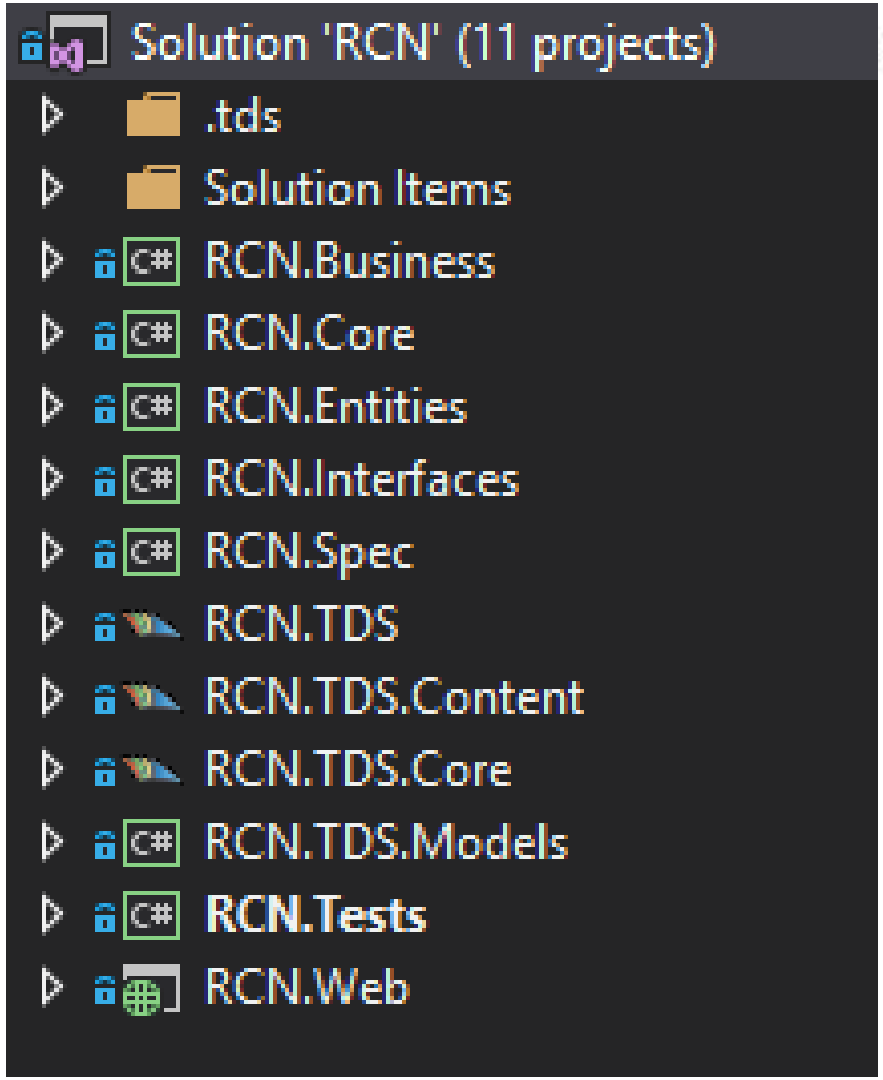
Align with the direction of the Industry

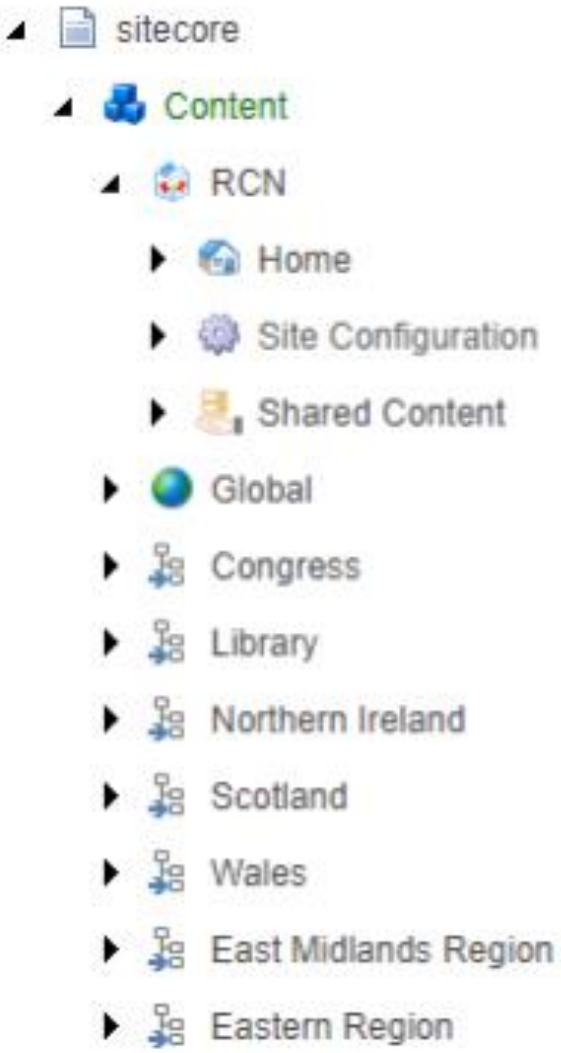
Improve as a developer

Clients probably don’t want to pay

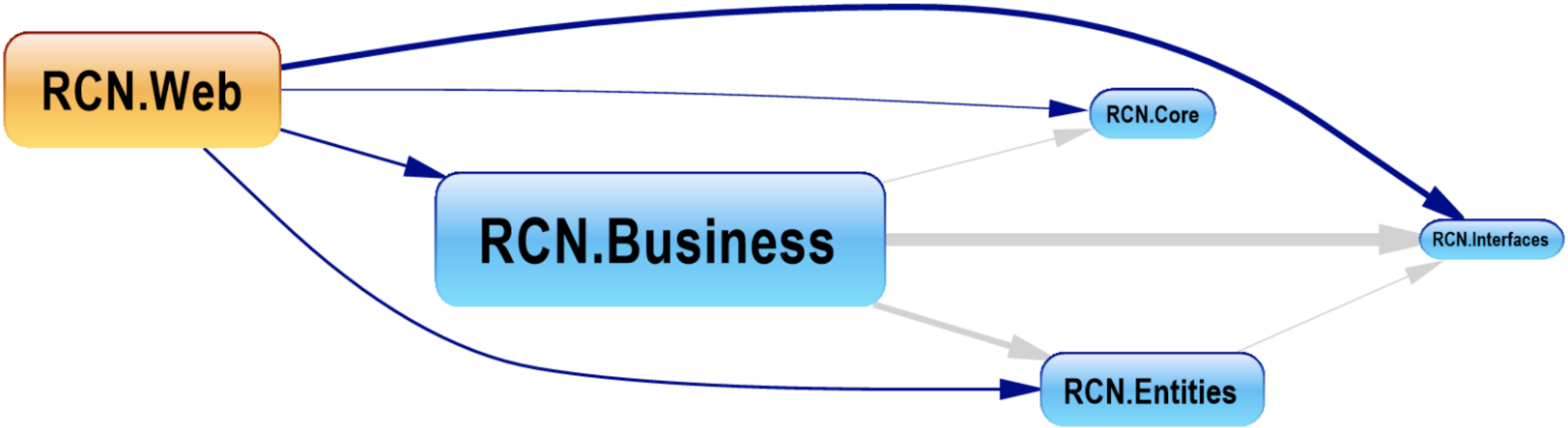
Undeveloping is fun!

# What it all looked like



# How the solution was structured



# Figuring out what's right for us

Development Team

RCN - Membership Organisation

and Trade Union

Varied audiences

Enterprise solution

Volumes of traffic

Delivering a progressive product

# Where we wanted to be

Harden the foundations

Remove limitations of N Tier

Enable future phases

Sort daily headaches

Use modern solutions to problems

But also…

# MAKE DEVELOPMENT GREAT AGAIN

# Before we started

Buy in

Robust branching strategy

Unit Test and Regression Test plans

A few Resharper licenses

Bravery!

# What we considered

What is right for us

SOLID principles

Design Patterns

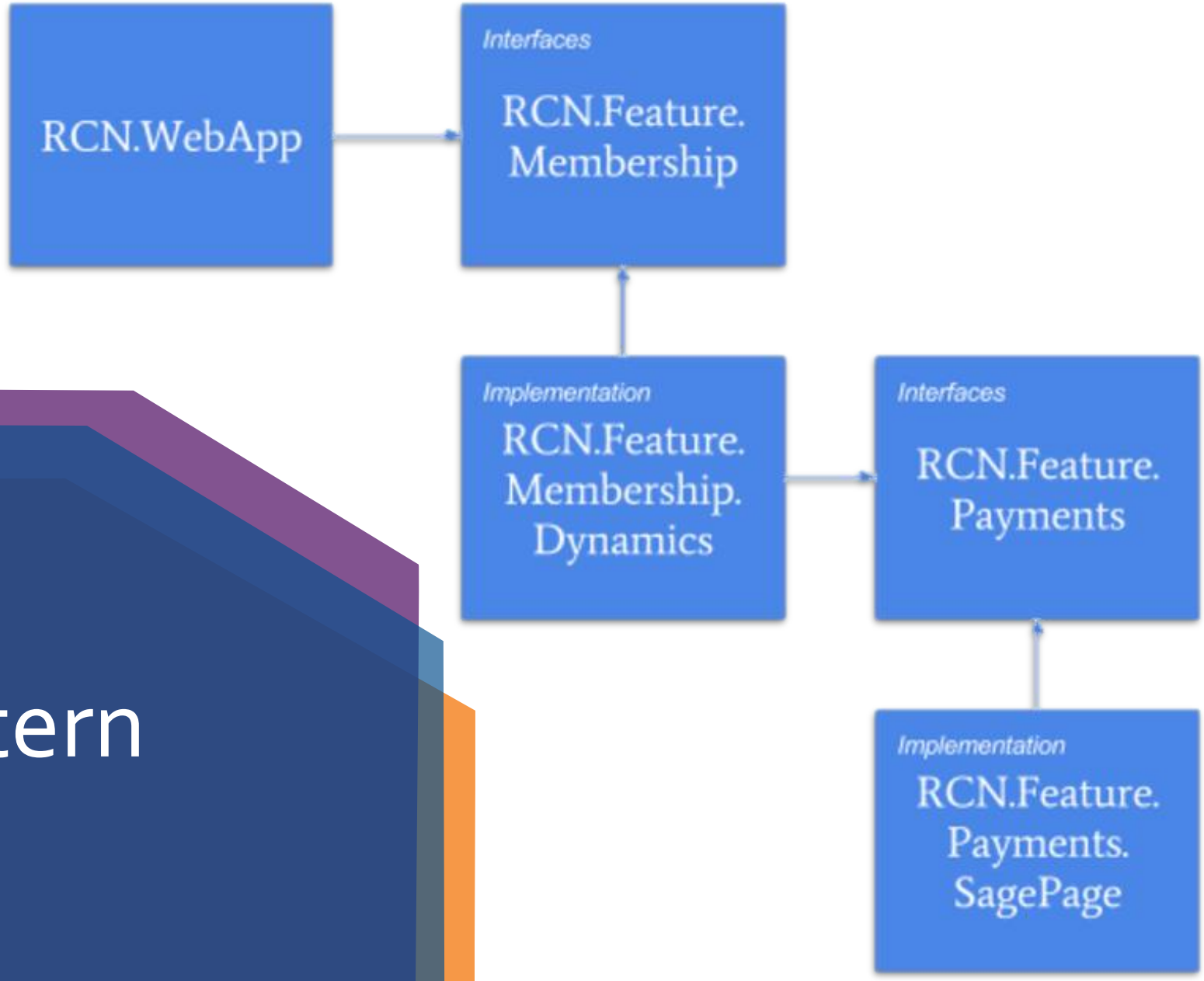
Modular Design

Development standards

Content strategy

Helix

# Stairway Pattern



# How we went about it

1. Chose technologies, principles and design patterns
2. Rewrote development guidelines
3. Completed a solution review
4. Audited the information architecture
5. Created a transformation backlog
6. Agreed an implementation approach

# How we paid our debts

Big Bang approach simply not possible

Phased approach

Features at a time

Boy Scout Rule

# How we tested our work

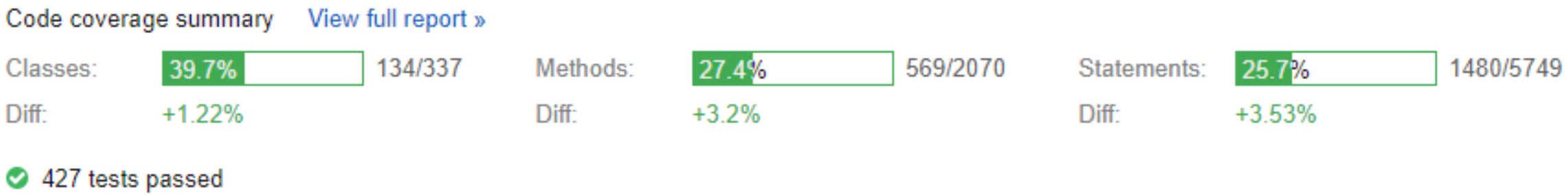


No Test or QA team!

CI server builds every commit to every branch

Unit Tests are run

Code Coverage reports are generated



# 77

Days of technical debt

# 14

Days of technical debt

# Amount of tech debt we've reduced

Technical Debt rating - D

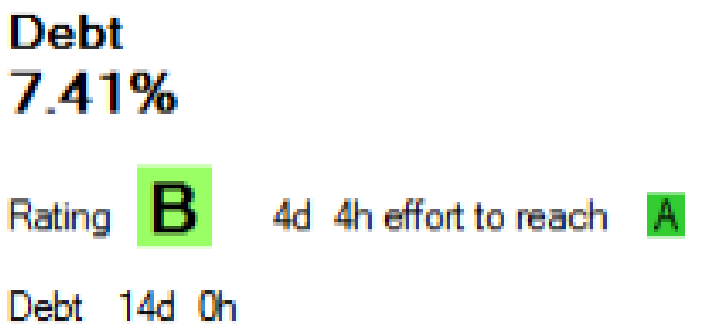
Technical Debt ratio - 20.83%

Days of debt - 77


Technical Debt rating - B

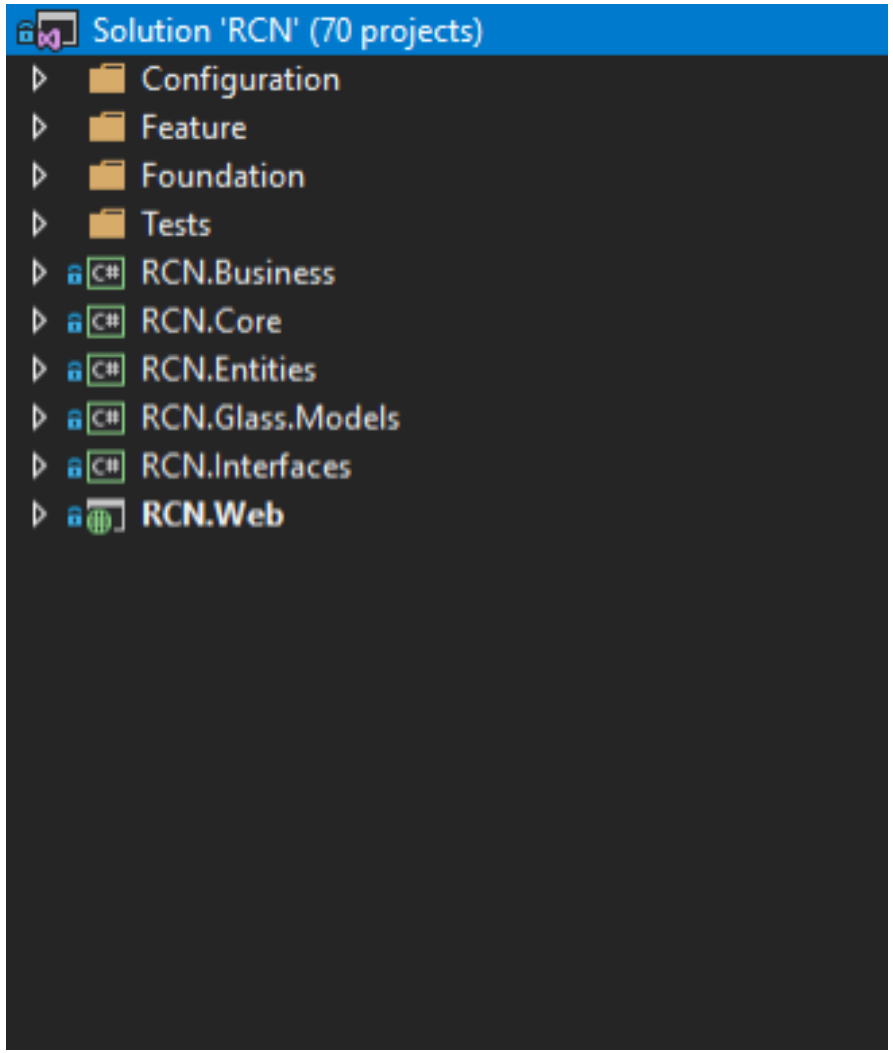
Technical Debt ratio - 7.41%

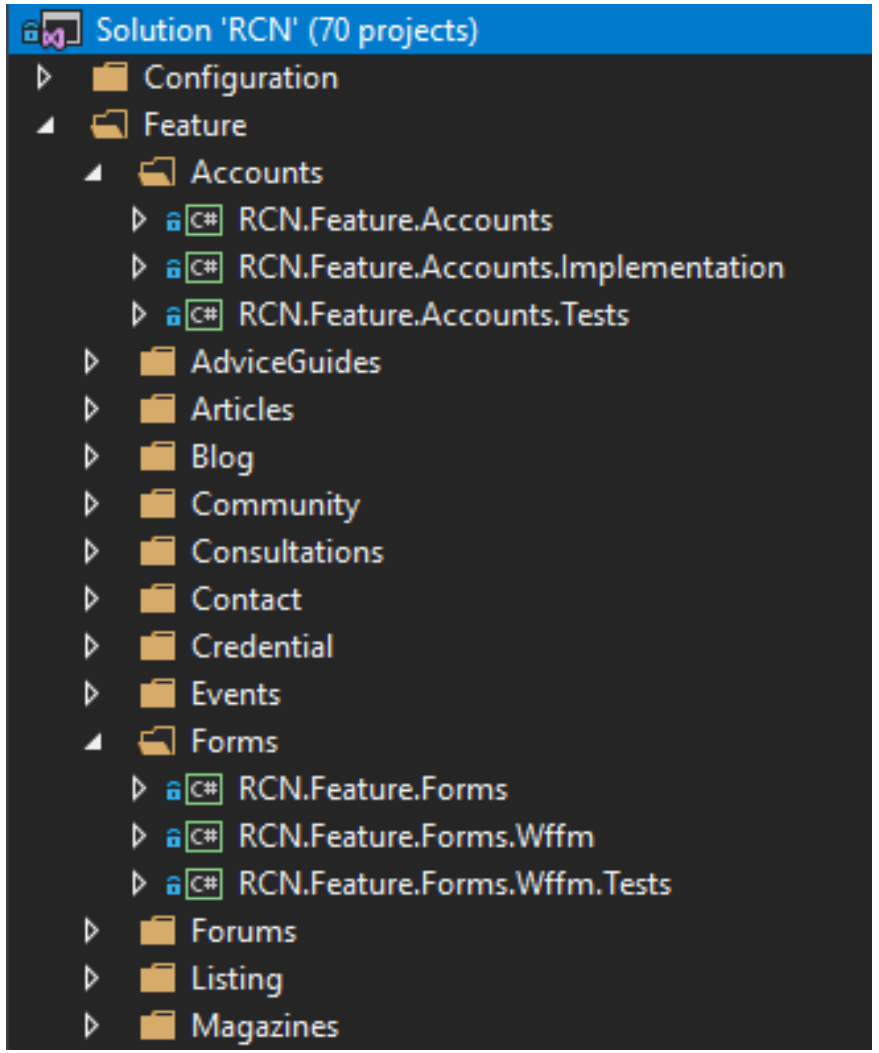
Days of debt - 14

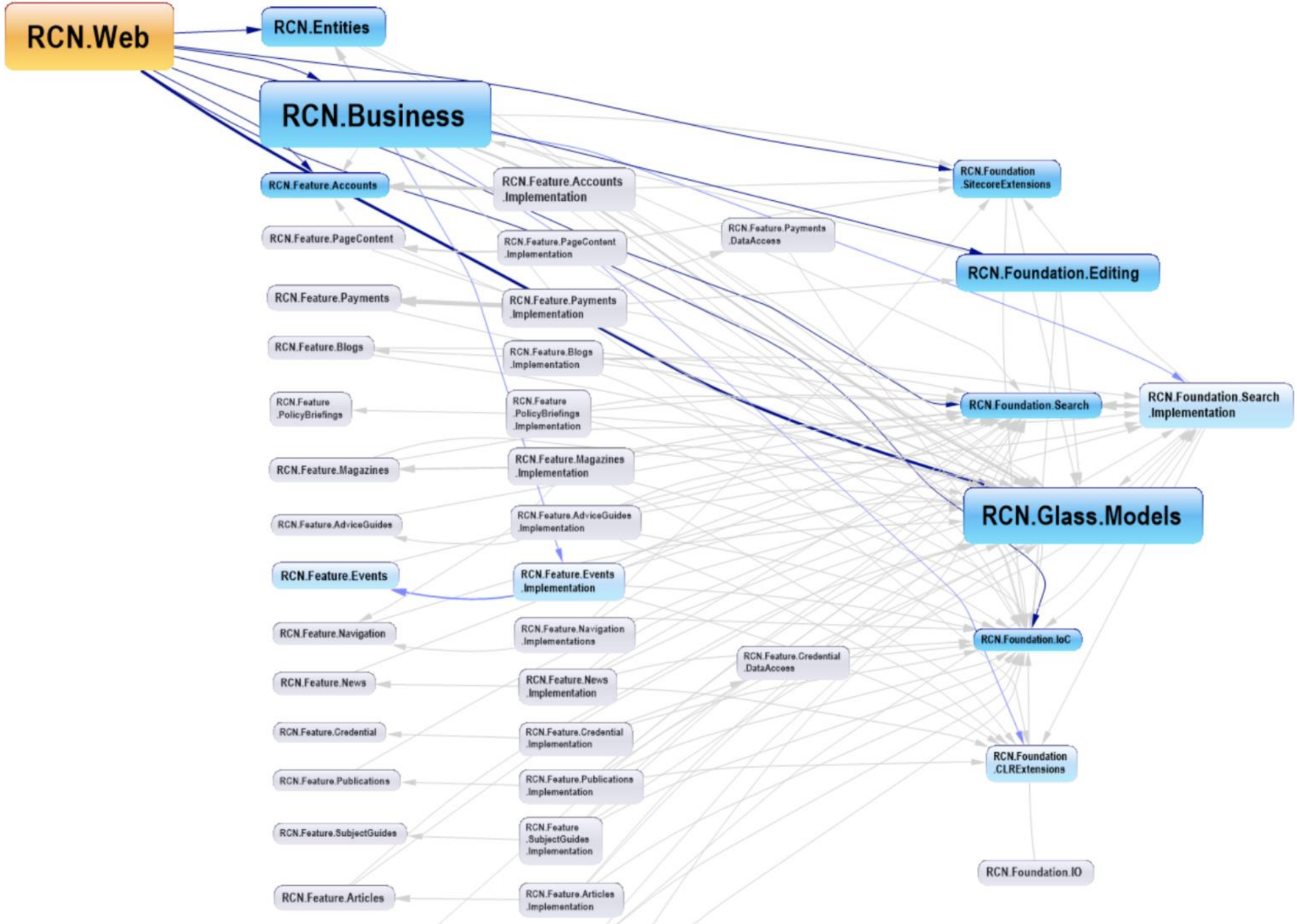


# So what it all looks like now

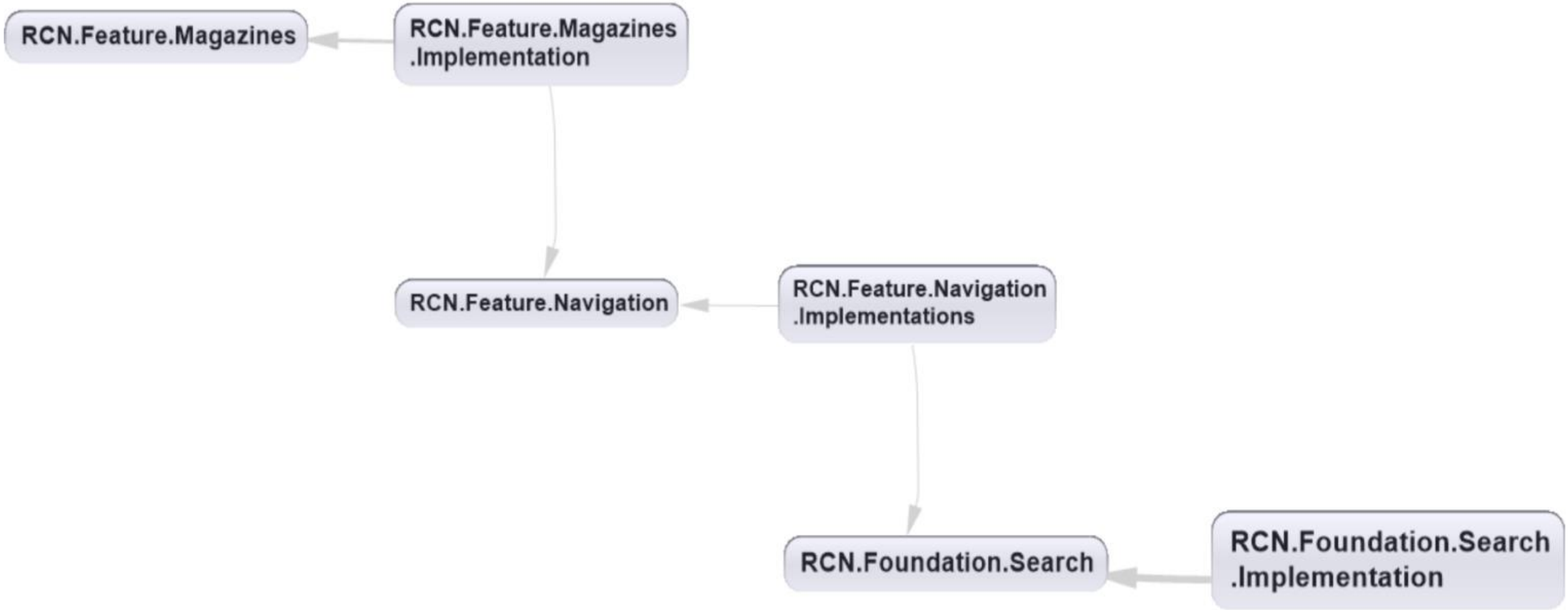





RCN.Web
RCN.Entities
RCN.Business
RCN.Feature.Accounts
RCN.Feature.Accounts .Implementation
RCN.Foundation .SitecoreExtensions
RCN.Feature.PageContent
RCN.Feature.PageContent .Implementation
RCN.Feature.Payments .DataAccess
RCN.Foundation.Editing
RCN.Feature.Payments
RCN.Feature.Payments .Implementation
RCN.Feature.Blogs
RCN.Feature.Blogs .Implementation
RCN.Feature .PolicyBriefings
RCN.Feature .PolicyBriefings .Implementation
RCN.Foundation.Search
RCN.Foundation.Search .Implementation
RCN.Feature.Magazines
RCN.Feature.Magazines .Implementation
RCN.Glass.Models
RCN.Feature.AdviceGuides
RCN.Feature.AdviceGuides .Implementation
RCN.Feature.Events
RCN.Feature.Events .Implementation
RCN.Feature.Navigation
RCN.Feature.Navigation .Implementations
RCN.Foundation.IoC
RCN.Feature.Credential .DataAccess
RCN.Feature.News
RCN.Feature.News .Implementation
RCN.Feature.Credential
RCN.Feature.Credential .Implementation
RCN.Foundation .CLRExtensions
RCN.Feature.Publications
RCN.Feature.Publications .Implementation
RCN.Feature.SubjectGuides
RCN.Feature .SubjectGuides .Implementation
RCN.Foundation.IO
RCN.Feature.Articles
RCN.Feature.Articles .Implementation

# Lets zoom in to a feature



# Our takeaways from doing this

Know where you're heading first

Don't go blindly following Helix

Be careful of pulling code weeds

Talk in words clients can understand

Sitecore Powershell Extensions saves (personal)lives

You might wanna check in those deployment packages…

# 3 months on - how has it helped us?

Handle far higher loads

Less tickets, more development

Juniors setting less fires

Faster implementation times

Easier communication with clients

Upgrading to 9 is more simple

But we're still not done...

# TL;DL - How do I solve my tech debt and move to a modern solution?

1. Get the thumbs up
2. Figure out what is right for you
3. Decide how you want the implementation to look in 12 months
4. Find approaches and principles that can help you get there
5. Review the solution against all this to build a backlog
6. Agree how to implement the changes with minimum risk

Now go break your site!










# Thank you!

Wes - @WesleyLomax
https://wesleylomax.co.uk

Jon - @ISlayTitans
https://jonathanrobbins.co.uk