Maintaining a high-traffic Sitecore application

and surviving to tell the story

Daniel Dumitrache

Technical Lead, Delete Agency

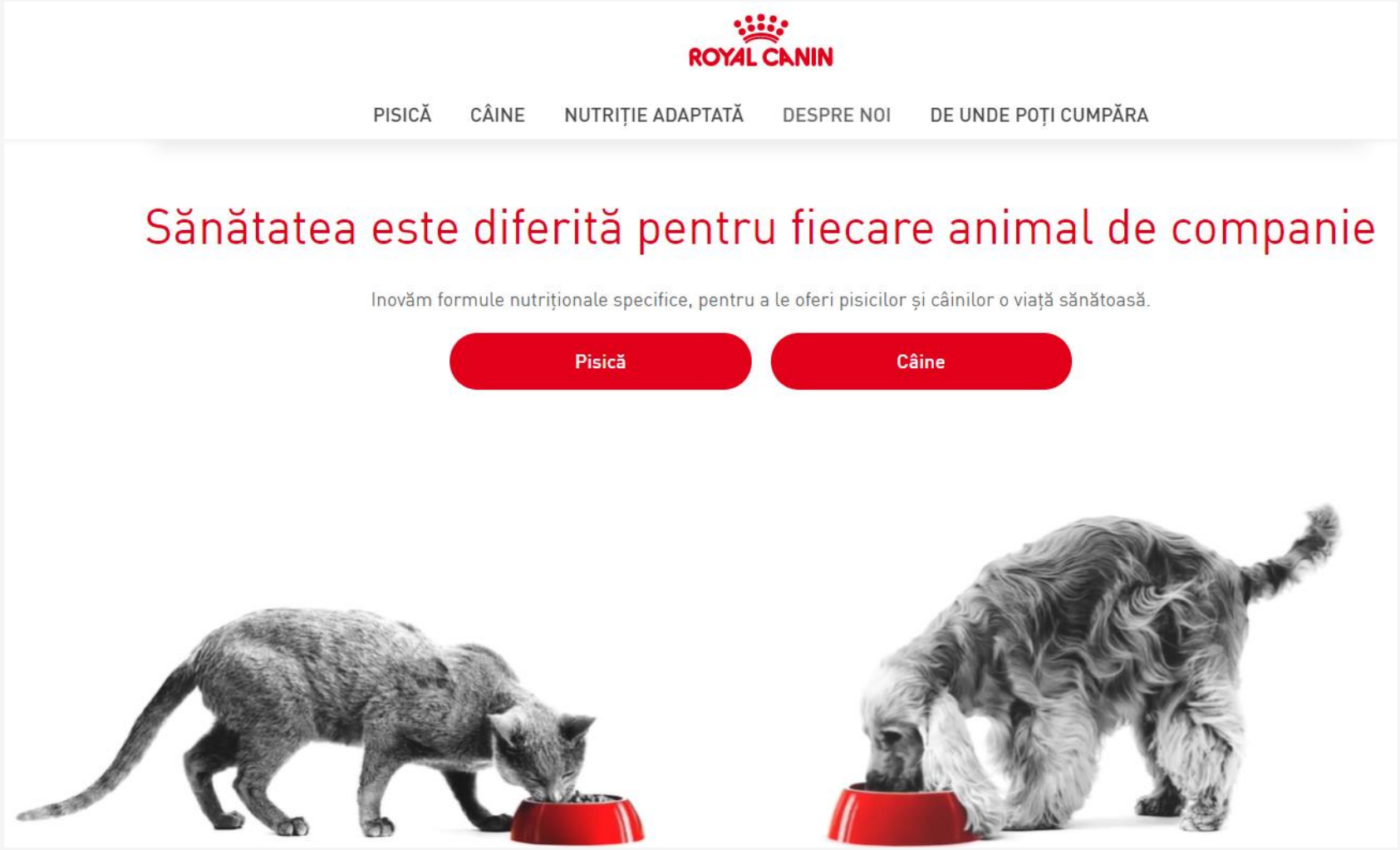
Our website

Multisite/multilanguage website promoting dog and cat food produced by Royal Canin.

49 different sites

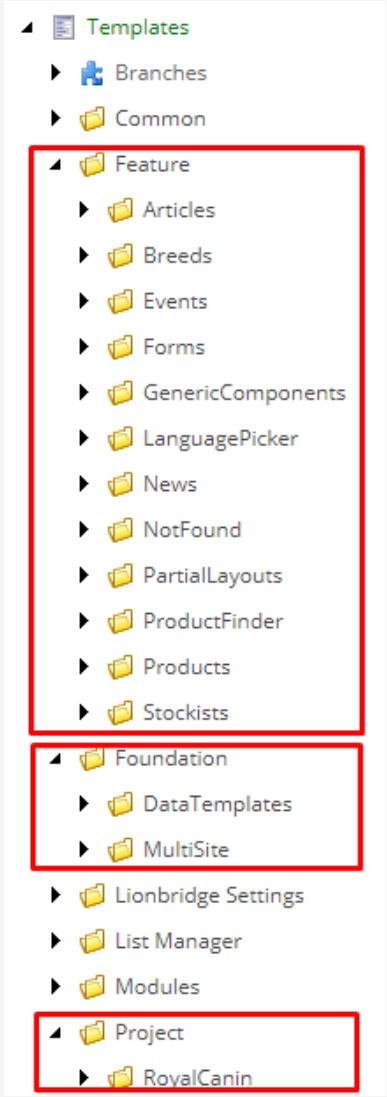
~20 million visitors per month from 49 different countries.

Google Analytics showing 300-800 concurrent users.

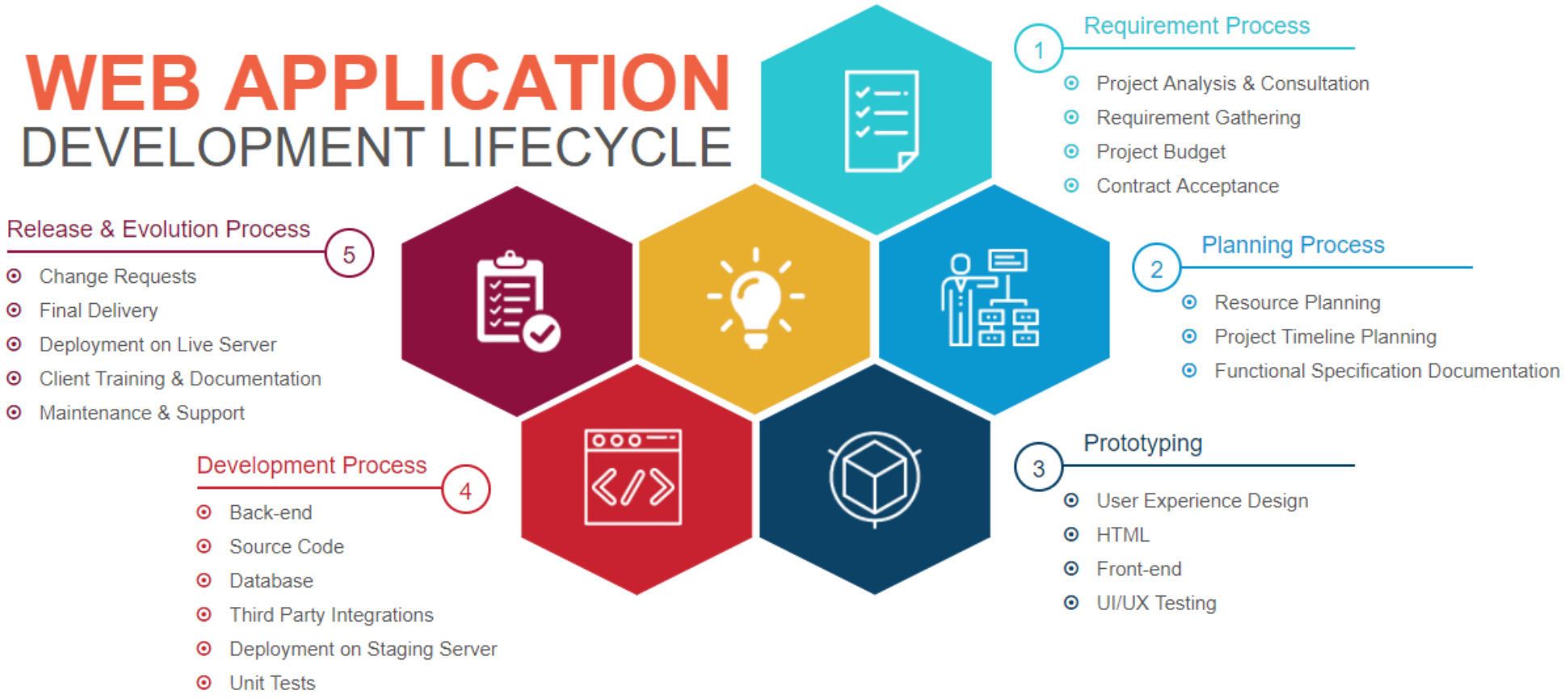


Solution and physical architecture

1. Based on Helix principles, Royal Canin website contains 3 levels of modules: Foundation, Feature and Project
2. Live website is an Azure PaaS app, based on XP Scaled topology (with modifications)
3. A Stage and QA environment mimics Live environment, with the same topology but lower resources
4. Dev environment is based on XP Single topology
5. Unicorn used for synchronizing content between environments



The lifecycle of a web application



Transition to long-term support

What we need to do

1. **Maintain website up 99.98% and within NFRs**
   NFRs 3-4 seconds response times on most pages
2. **Add new features**
   New features, change requests, bug fixes
3. **Offer support to Royal Canin staff**
   Support through Zendesk to RoyalCanin staff

Problems

1. **Slow website**
   6-7 seconds average response times
2. **Downtime during deployments**
   Manual deployments with maintenance page
3. **Website crashing during peak times**
   60+ seconds response times during work-day

When things go bad

Incident Reference	**Issue reported by Delete, and later logged by Home #7893**
Date Incident Occurred	**11/11/2019**
Date Incident Resolved	**11/11/2019**
Severity	**Critical**
Incident Duration	**Slow response time: 10:48 - 15:07 (4h19min)** **Unavailable site/Maintenance Page (except US): 15:07- 16:50 (1h43min)**
Incident Impact	High response time and sporadically unavailable pages recorded on the Royal Canin HUB, Maintenance Page being set during the CD restarts and leading to site unavailability for all markets except the United States.
Incident Root Cause	Recent publishing activities done in the view of new markets roll out (+6 markets in the last week) and the recent traffic increase (10x) led to high DTU consumption on the WebDB, reflected in a high response time of the website.

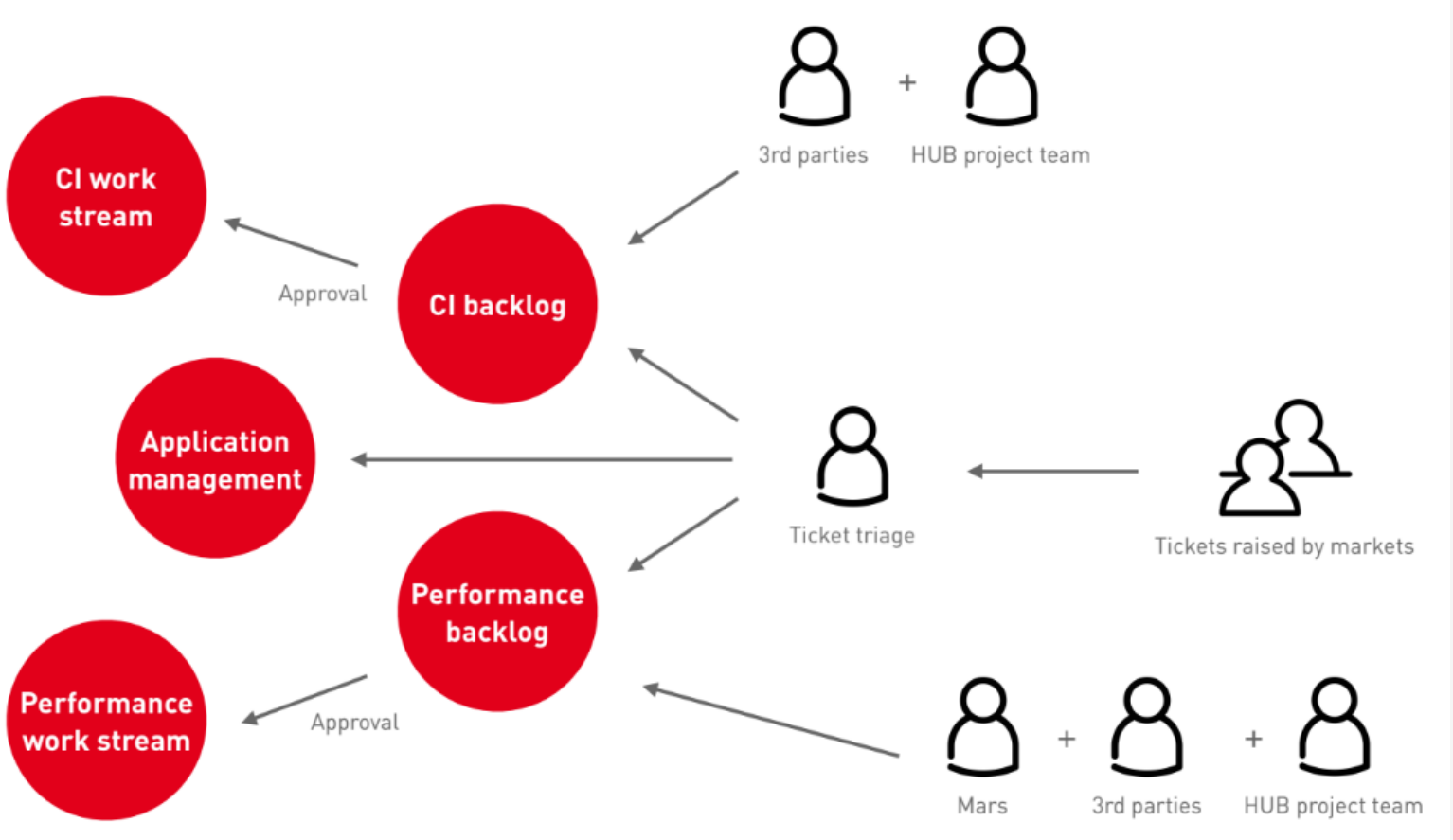
Splitting the team in multiple workstreams

3 workstreams:

1. Application Management -> handles support requests, deployments, application monitoring and on-call support

2. Continuous Improvement -> works in 2-week sprints, deliverables are new features and bug fixes

3. Performance -> focus on website speed, reliability, efficient use of server resources and security



Improving the website – Performance Workstream

Objectives

1. No downtime at deployments or anytime
2. Improve website stability
3. Improve website speed to fall within NFRs
4. Monitoring and reporting
5. Optimize AM processes
6. Ensure website is secure

Things inherited

1. Vertical scaling on CD servers to maximum (I3 configuration, 14GB RAM)
2. CD using 4 instances in a load balancer
3. Web database temporary scaled up to 400 DTU

1. No downtime deployments

1 Azure deployment slots

One of the best features of Azure App Services.

Red/black deployment between 2 instances.

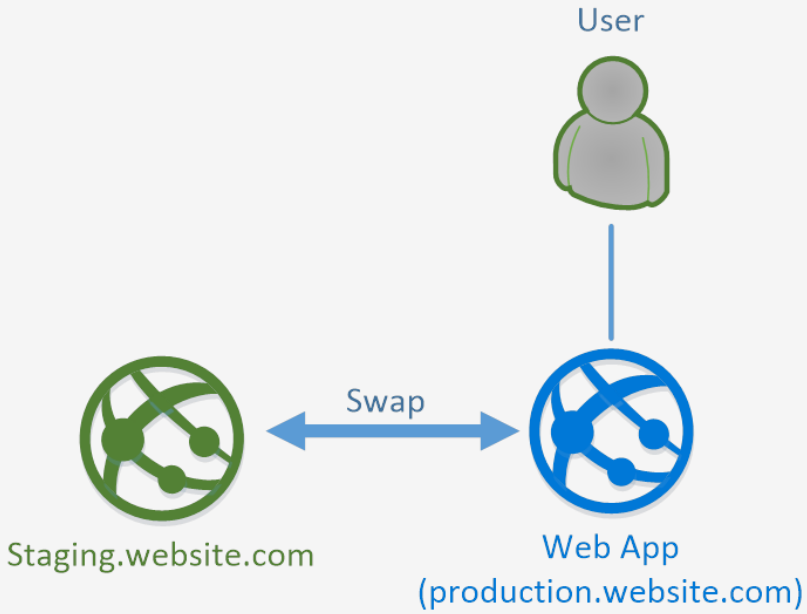
Able to rollback deployment slot if necessary

2 Enable Local Cache

Protect application against maintenance and upgrades in Azure

Each worker instance gets a copy of the website contents

Needs individual instance restarts after deployment



2. Improve website stability

1 Sitecore caches optimization

Some of the caches refreshed every 2 minutes, adding strain to Master/Web databases and increasing response times

2 SQL indexes

Azure suggests some indexes – and some of them might help

3 Fix logged exceptions

Scan logs in application insights for recurring exceptions

4 Monitor servers' usage

Look for patterns in servers' usage and fix problems proactively

5 Use 3rd parties' support

Ask Azure, Sitecore, Lionbridge anytime an anomaly happens why

6 Disable potentially harmful features

Use well-defined roles

Disable Publish Website

Restrict potentially damaging PowerShell scripts from running Live

4 Automated testing

Automated regression tests, load tests on new features

3. Improve website speed – client side

Things to look for

1. Minify and compress CSS/JS/HTML
2. Optimize image sizes and formats
3. Use srcset for different resolutions
4. Lazy loading for images
5. Enable HTTP/2
6. Remove unused resources

Monitoring tools

1. Google Page Speed Insights
2. Lighthouse
3. WebPageTest
4. Google Search Console

3. Improve website speed – server side

Objectives

1. Reduce database roundtrips
2. Cache common operations
3. Use CDN for static resources
4. Use a distributed application cache
5. Edge caching
6. Publishing Service

Tools

1. Google Analytics
2. Application Insights
3. Sitecore Debug Mode
4. Sitecore Pipeline Profiler
5. Memory Profiler
6. Local/Remote Debugging

3. Improve website speed – server side

April response times:

OPERATION NAME	DURATION (A...	COUNT	PIN
Overall	**1.47 sec**	**39.23M**	
GET stockists/filter	5.07 sec	485.02k	
GET Sitecore/Index	2.43 sec	19.32M	
GET products/searchproducts	2.26 sec	788.88k	
GET royalcanin/predictive	2.11 sec	587.53k	
GET breeds/findbykeyword	2.09 sec	389.26k	
GET products/findbykeyword	1.39 sec	281.23k	

May response times:

OPERATION NAME	DURATION (A...	COUNT	PIN
Overall	**996 ms**	**35.59M**	
GET Sitecore/Index	1.58 sec	20.43M	
GET royalcanin/predictive	828 ms	610.29k	
GET breeds/findbykeyword	763 ms	408.98k	
GET products/searchproducts	629 ms	841.13k	
GET products/findbykeyword	502 ms	287.78k	
GET stockists/filter	432 ms	649.32k	

4. Monitoring and reporting

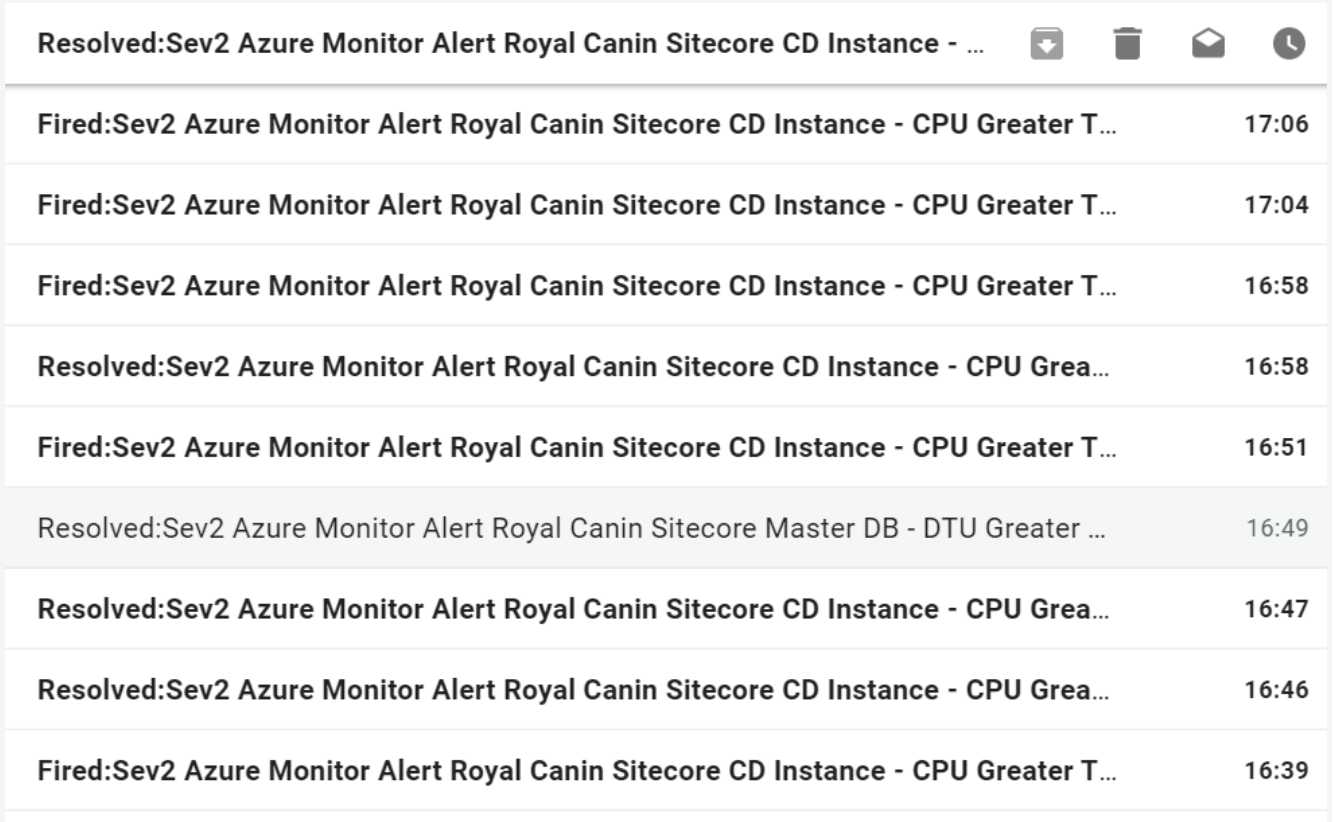
3 workstreams:

1. Azure Alerts -> 20-second response times, memory/CPU usage over 75%; too many alerts can be harmful

2. Downtime monitoring -> Pingdom, StatusCake, Service Desk

3. Ad-hoc monitoring -> logs

4. By-monthly reports -> website response times, resource usage, infrastructure costs, potential issues



5. Optimizing Application Management Team (AM) Work

As the application grew, the number of tickets increased. Repetitive work needed to be automated

#markets live	1				2		5	11	15	30	38	47	48	49
Month	Jan 2019	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan 2020	Feb 2020
Logged	1		1	5	2	10	12	33	116	142	172	152	144	137
Solved									55	115	135	140	206	175
Avg. cycle time (business days)[1]									10	12	15	26	34	29
Backlog (estimated)									125	152	189	201	139	103

5. Optimizing Application Management Team (AM) Work

What the AM team does

1. Application Monitoring (<5%)
2. Regular/ad-hoc deployments (10%)
3. Support for Royal Canin staff through Zendesk (80% -85%)
4. On-call support (<5%)

Things to improve

1. Application Monitoring -. Set up alerts and monitoring dashboards
2. Regular deployments -> do automated instead of manual deployments
3. Zendesk support -> 30% of time is taken by data imports from 3rd parties
4. On-call support -> increase application stability

6. Ensure website is secure

1 Follow OWASP top 10 web application security risks

Most are fixed by design and taken care of by the framework

Review 3rd party libraries before using

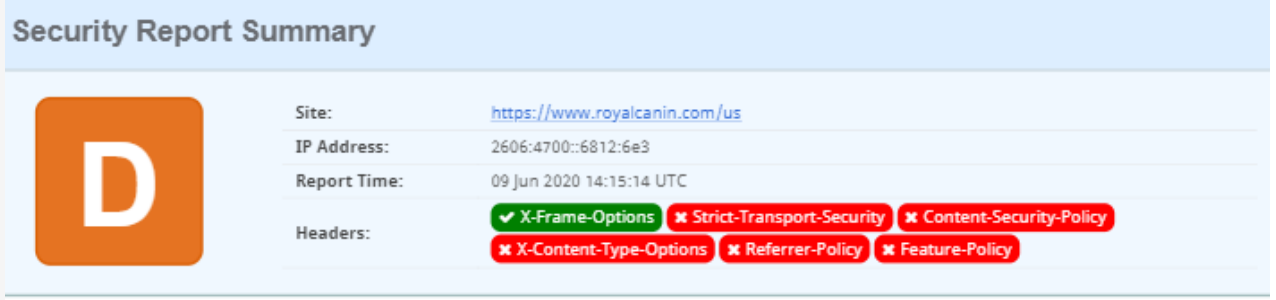
2 External audits

Offers more trust than internal audits

Purpose-trained people for security pen-testing

3 Sitecore Security Bulletins

Periodic updates on vulnerabilities



Improving the website – where we are now

Done

1. No downtime deployments
2. Automated deployments (qa/stage)
3. Website speed <3seconds
4. Alerts, monitoring and reports ongoing
5. Website secure

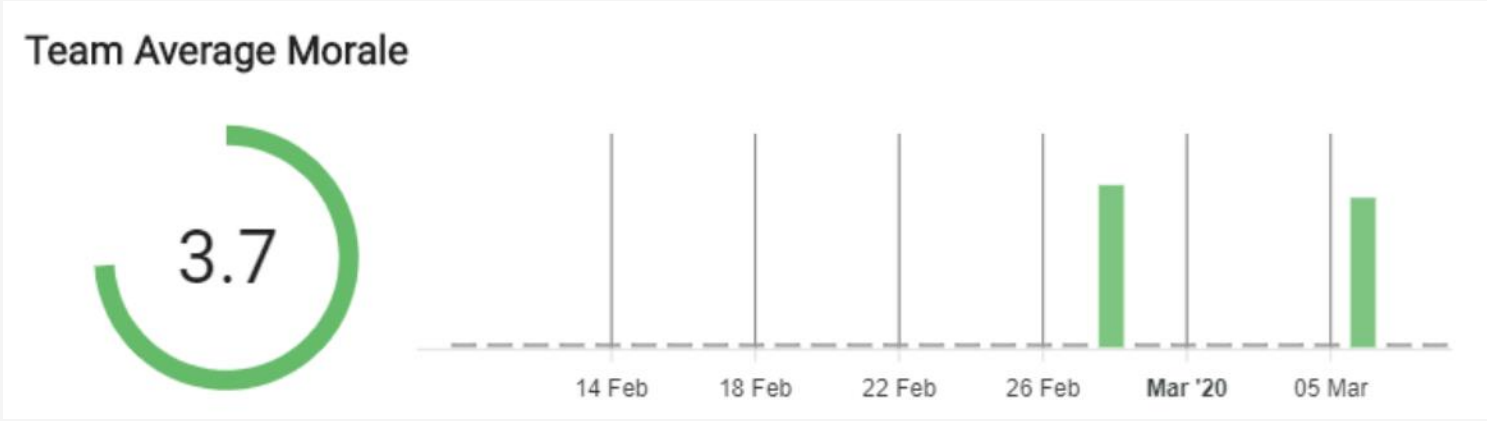
Things to do

1. Automated deployments (prod)
2. Mobile speed improvements
3. Automate or delegate 3rd party imports
4. Use Redis cache
5. Automated regression testing

Making things work

How it works:

1. Team communication is important -> between AM and Dev teams, between Royal Canin and Delete
2. Team rotations -> Dev and AM members switch places or work on both streams, so they know code and issues
3. Feedback -> listen to feedback from the team and client
4. New team members -> fresh ideas are good



Questions?

Contact me at daniel.dumitrache@deleteagency.com

Visit our website – https://deleteagency.com

Delete™