
Modern Apps acceleration with
CAST HIGHLIGHT
Retail Case Study

Application Portfolio Analysis

Health & Cloud Migration Assessment

Key Metrics & Methodology Definitions

Key Metric	Description	Direct Interpretation	Business Impact
Software Resiliency	Measure the robustness and how bullet-proof is the Software against production failure	Reflects presence of code patterns that may comprise vulnerability of the software	Customer Satisfaction Customer Confidence / Loyalty Opportunities & Revenue
Software Agility	Measure to indicate the easiness of a development team to understand and maintain an application	Reflects absence of embedded documentation and code readability good practices	Maintenance Cost Transferability
Software Elegance	Measures the ability to deliver software value with less code complexity	Indicates decreased quality in code, resulting in higher defects that become costly to fix	Time to Market Innovation
Cloud Readiness	Measure of software and organization characteristics to speed PaaS migration	Significant number of roadblocks found that could slow down a Cloud migration	Opportunity to reduce cost, increase elasticity and embrace innovation

Resiliency	Agility	Elegance
High > 84.0	High > 71.0	High > 78.0
Medium > 62.0	Medium > 56.0	Medium > 55.0
Low < 62.0	Low < 56.0	Low < 55.0

June 2018

Application Portfolio Analysis – a Continuous Process

Application Diagnostics

Rapid Portfolio Analysis meant to provide objective, repeatable and cost/time efficient assessment of large, decentralized as well as complex IT portfolios.

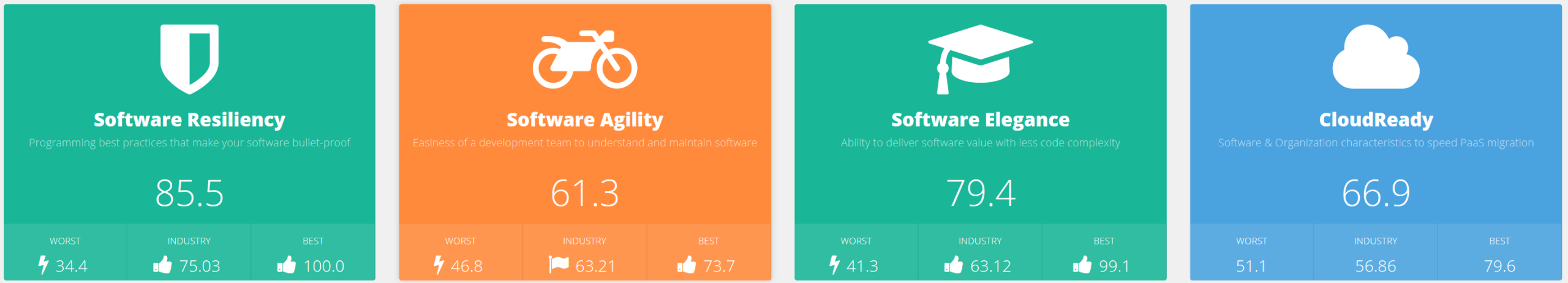
Get an Objective Baseline	Segment to Prioritize	Build Recommendations & Set Objectives	Continuous Improvement
Transparency - Truth	Multi criteria analysis – Portfolio Level	Multi criteria analysis – Application Level & Actionable Outcomes	Trending – Automation - Extension

June 2018

CAST Highlight Report

Portfolio Baselining

Application Portfolio Analysis - Baseline



14
applications

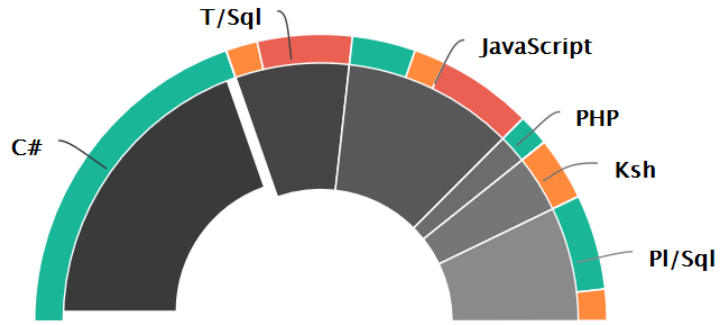
797.8k
lines of code

18.3
full-time equivalents

10k
back-fired function points

June 2018

Application Portfolio Analysis – Technology Mix



Search:

Technology	LOC	Apps	SR	SA	SE
C#	330k LOC	11	96.97	63.85	86.92
T/Sql	228k LOC	4	51.31	55.48	80.96
JavaScript	206k LOC	6	64.83	51.52	55.44
PHP	30k LOC	1	94.72	74.04	86.98
Ksh	3k LOC	2	68.31	66.78	96.08
Pl/Sql	949 LOC	4	89.45	74.28	98.36

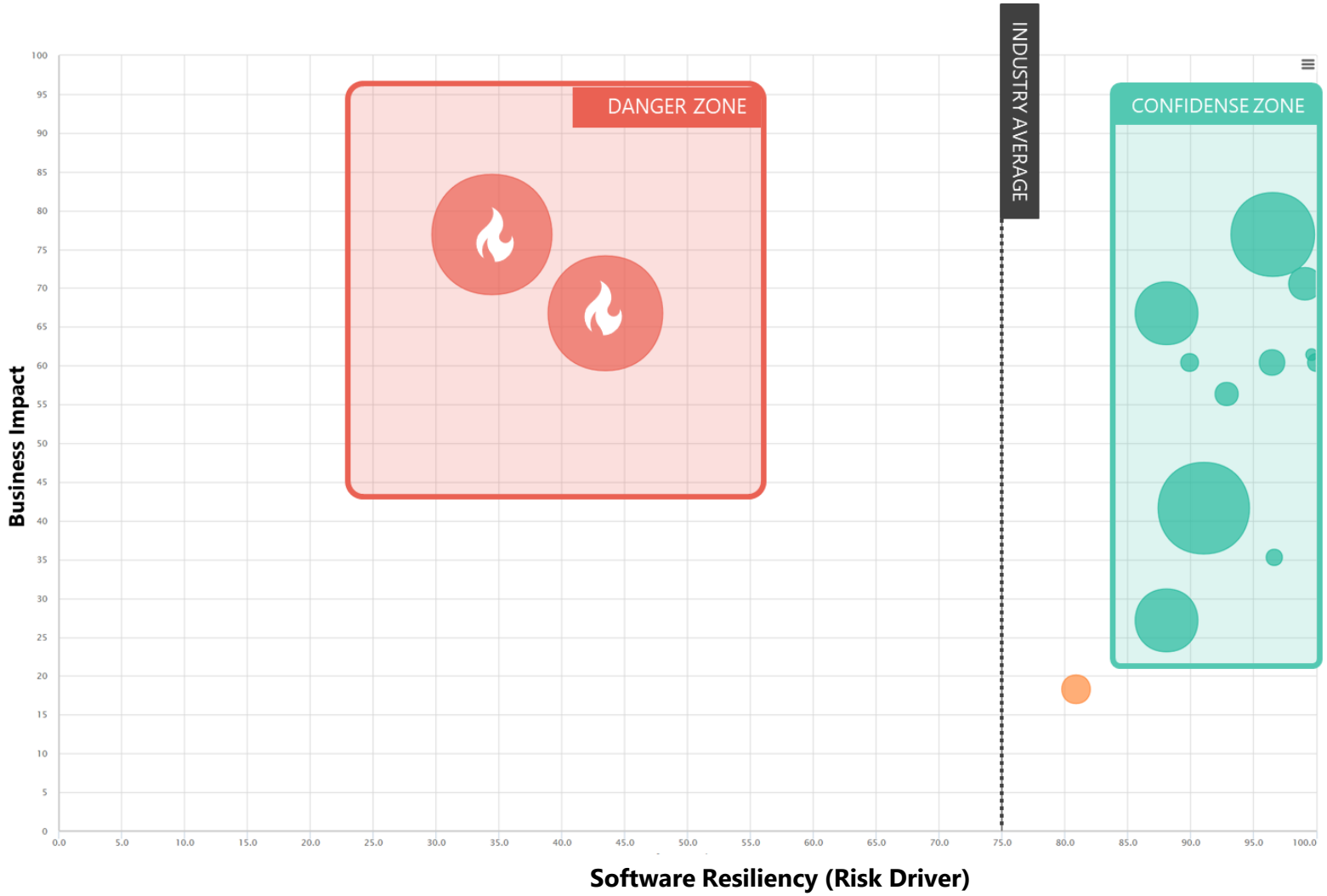
Application Portfolio Analysis – Demographics

Key Insights	Description	Comments
Evolution	Number of major releases delivered over the last 12 months	**Fast-evolving applications** - 11 apps get more than 12 - 3 apps with less than 3
User Exposition	What is the approximate number of end users	**Great exposition** - 12 apps serve more than 2,500 users
Impact of Application Failure	Could failure lead : - to disruption - Loss of revenue or business opportunity - Harm the company's public image - Loss of customer confidence	**High Business Impact** - 11 applications own High Impact

CAST Highlight Report

Portfolio Health Analysis

Business Impact Vs Software Resiliency



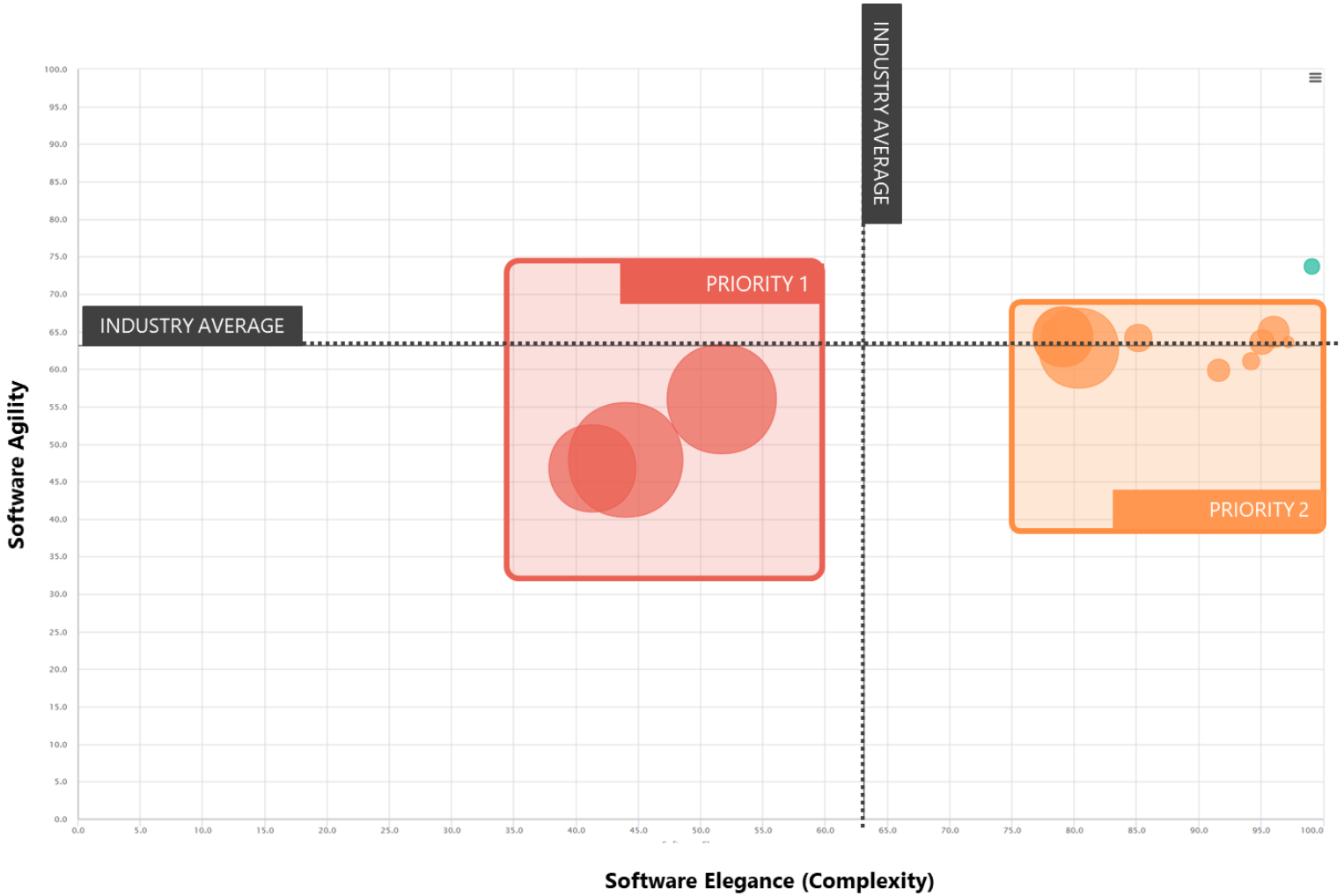
DANGER ZONE

SGBDNL - EBPSQL
SGBDNL - PhpSgdOnLine

CONFIDENCE ZONE

SGBDNL - FeatureAdvantage
SGBDNL - MailQueue
SGBDNL - ERP
SGBDNL - Reporting
SGBDBNL - API
SGBDBNL – Mccm
SGBDBNL – Loyalty
SGBDBNL – WCF – EBusiness Platform
SGBDBNL – Solr
SGBDBNL – CDN
SGBDBNL - CMS

Software Agility Vs Software Elegance



PRIORITY 1

SGBDNL - EBPSQL
SGBDNL - PhpSgdOnLine
SGBDBNL – WCF – EBusiness Platform

PRIORITY 2

SGBDNL - FeatureAdvantage
SGBDNL - MailQueue
SGBDNL - ERP
SGBDBNL - API
SGBDBNL – Mccm
SGBDBNL – Loyalty
SGBDBNL – WCF – EBusiness Platform
SGBDBNL – Solr
SGBDBNL – CDN
SGBDBNL - CMS

Portfolio Health Assessment

Name	Category	SR	SA	SE	CloudReady	Roadblocks	BI	ROAR	Total FTE	$ Tech.Debt/LOC	BFP	LOC
SGBDNL - EBPSQL	○	34.42	47.90	43.96	72.43	0	77	47.9	4.00 FTE	3 $/LOC	4k	228k LOC
SGBDNL - PhpSgdOnline	○	43.45	56.01	51.67	79.64	0	67	35.8	3.00 FTE	3 $/LOC	2k	204k LOC
RedMine	□	80.87	64.15	85.12	51.10	0	18	3.6	0.00 FTE	6 $/LOC	47	7k LOC
SGBDNL - CDN	○	88.07	64.31	79.07	65.55	44	67	10.8	0.10 FTE	8 $/LOC	578	52k LOC
SGBDNL - CMS	○	88.07	64.31	79.07	59.96	44	27	4.4	0.10 FTE	8 $/LOC	578	52k LOC
SGBDNL - Solr	○	89.90	65.33	78.08	65.38	20	60	9.0	0.50 FTE	6 $/LOC	33	3k LOC
SGBDNL - WCF - EBusiness Platform	○	91.03	46.77	41.29	60.84	199	42	9.7	1.00 FTE	3 $/LOC	1k	123k LOC
SGBDNL - Loyalty	○	92.85	59.84	91.57	68.23	14	56	6.0	1.00 FTE	8 $/LOC	55	5k LOC
SGBDNL - Mccm	○	96.46	63.60	95.07	68.66	40	60	4.3	0.20 FTE	9 $/LOC	63	6k LOC
SGBDBNL - API	○	96.52	62.78	80.34	63.91	129	77	7.8	4.00 FTE	8 $/LOC	1k	100k LOC
SGBDNL - Reporting	○	96.63	73.69	99.07	75.21	12	35	1.8	0.10 FTE	11 $/LOC	31	3k LOC
SGBDNL - ERP	○	99.07	64.97	95.99	63.06	33	71	3.5	4.00 FTE	10 $/LOC	114	10k LOC
SGBDNL - MailQueue	○	99.60	63.60	97.17	72.56	26	61	2.8	0.20 FTE	11 $/LOC	26	2k LOC
SGBDNL - FeatureAdvantage	○	99.99	61.05	94.18	69.99	11	60	3.1	0.10 FTE	10 $/LOC	33	3k LOC

Common Vulnerabilities & Exposures via Frameworks

CVEs have been detected via Frameworks used by 4 different apps

6	17	1	0
CRITICAL SEVERITY	HIGH SEVERITY	MEDIUM SEVERITY	LOW SEVERITY

SGBDNL – Loyalty (6)
Swashbuckle.AspNetCore 1.0.0
System.ServiceModel.NetTcp 4.3.0
Microsoft.EntityFrameworkCore.SqlServer 1.1.2

SGBDNL – API (1)
Microsoft.Owin.Hosting 3.0.1.0

SGBDNL – ERP (2)
Microsoft.Owin.Hosting 3.0.1.0
Microsoft.Owin.Hosting 3.1.0.0

SGBDNL – Loyalty (13)
System.ServiceModel.NetTcp 4.3.0
Microsoft.EntityFrameworkCore.SqlServer 1.1.2
(...)

SGBDNL – Mccm (1)
Microsoft.Owin.Hosting 3.1.0.0

SGBDNL – Loyalty (1)
Microsoft.EntityFrameworkCore.SqlServer 1.1.2

June 2018

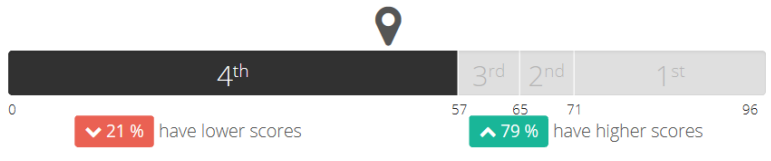
SGBDNL-EBPSQL Example – Code Insights

34.2
Software Resiliency

Improvement Opportunities	Frequency
The code contains too many functions and procedures doing an Insert, Update, Delete, Create Table or Select without including error management	40.39 %
The code contains too many functions and procedures doing an Insert, Update or Delete without managing a transaction	22.96
RETURN should be used to terminate a database query or a stored procedure	17.49 %
The code contains too many subqueries	7.91 %
The code contains too many uses of GROUP BY ORDER	4.86 %
The code contains too many tables without primary keys	1.73 %
The code contains too many queries on too many tables in the same request	1.44 %

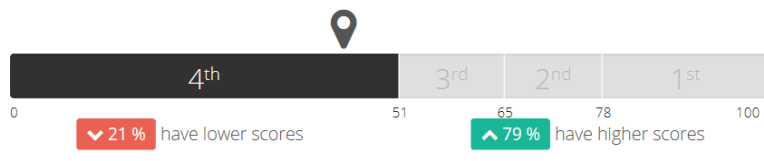
47.8
Software Agility

Improvement Opportunities	Frequency
The code has a low code/comment interleaving rate	27.62
The code contains too many functions and procedures parameters that are not commented	24.16
Ensure that your main SQL-related code artifacts are documented.	17.40 %
The code contains too many object qualifications with missing owner's specifications	9.87 %
A line of code shouldn't be too long to help readability.	8.48 %
The code contains too many insert instructions that do not specify the columns list of the table	6.85 %
The code contains too many WHERE clauses having several simple expressions per line	2.69 %
Code should be stripped out of suspicious comments (e.g. todo, tbd, tbc, etc.)	1.55 %
Code indentation, line, and character encoding formats should be consistent.	1.37 %



43.8
Software Elegance

Improvement Opportunities	Frequency
The code contains too many DDL and DML interleaving	42.92 %
The code contains too many queries on too many columns	17.80 %
Bulky files are complex to work with.	17.03 %
The code contains too many functions and procedures having too many parameters	12.85 %
The code contains too many functions, procedures and triggers having high cyclomatic complexity	9.39 %

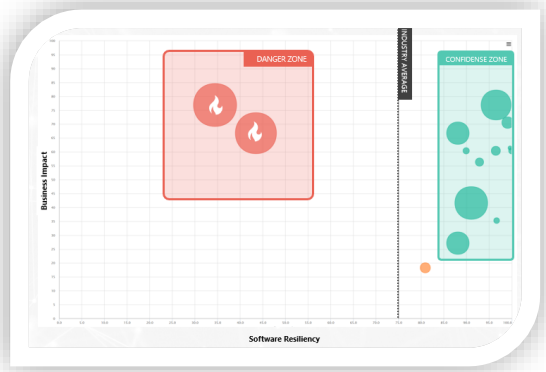


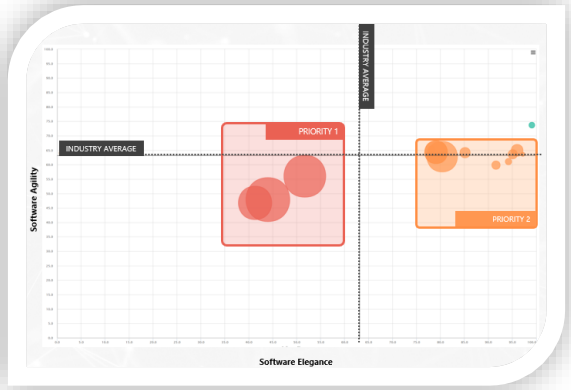
ROAR INDEX	SOFT. MAINTENANCE	CODE SIZE	BACKFIRED FPS
47.9	1.40 FTE	228k LoC	4k

TOTAL FTE	FILES	CONTRIBUTORS	TECHNICAL DEBT
4.00 FTE	3k	6	607k

June 2018

Software Health Analysis | **INSIGHTS**





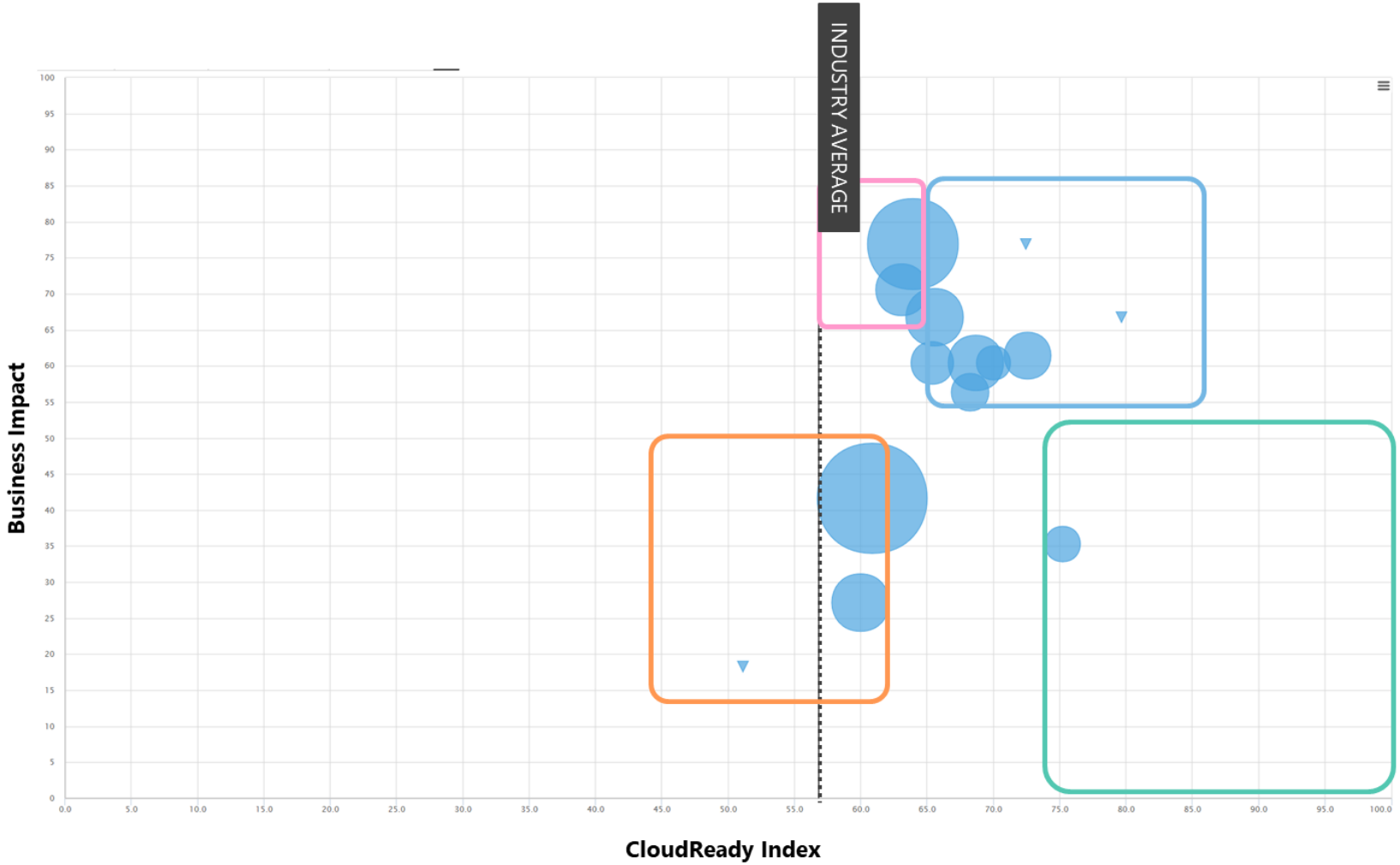
Synthesis

- EBPSQL and phpSgdOnLine are 2 Business Critical Application which are exposed to **production risk** and get a **complexity** above the average.
- Software Agility is globally average which can impact both **maintainability** & **transferability** of the applications.
- Amongst these 14 applications some used frameworks lead to **24** potential threats known as **Common Vulnerabilities & Exposures** (CVEs). Loyalty alone, owns 20 CVEs including 6 with critical severity.
- The **Code Insights** should be leveraged to **improve the health of applications** with the greatest ROAR (Ranking of Application Risks). These findings represent a concrete opportunity to transition Software Resiliency, Agility & Elegance scores from 4th Quartile to at least the 3rd Quartile. Therefore, avoiding disruption and hidden cost & time due to difficulties in maintaining & changing the Software.
- **Recommendations**:
  - Quickly check the detected CVEs to avoid security breaches
  - Segments applications to reprioritize and apply a continuous assessment to track progress over time

CAST Highlight Report

App Migration & Modernization

Cloud Migration Segments | Business Impact



QUICK-WINS

SGBDNL - Reporting

CORE CLOUD

SGBDNL - CDN
SGBDNL – EBPSQL
SGBDNL – FeatureAdvantage
SGBDNL – Loyalty
SGBDNL – MailQueue
SGBDNL – Mccm
SGBDNL – PhpSgqOnline
SGBDNL - Soir

Long-Term Bets

SGBDNL – API
SGBDNL - ERP

Pursue Later

SGBDNL – CMS
SGBDNL – WCF – EBusiness Platform
RedMine

Cloud Migration Segments | Roadblocks

Name	Category	SR	SA	SE	CloudReady	Roadblocks	BI
SGBDNL - WCF - EBusiness Platform	○	91.03	46.77	41.29	60.84	199	42
SGBDBNL - API	○	96.52	62.78	80.34	63.91	129	77
SGBDNL - CDN	○	88.07	64.31	79.07	65.55	44	67
SGBDNL - CMS	○	88.07	64.31	79.07	59.96	44	27
SGBDNL - Mccm	○	96.46	63.60	95.07	68.66	40	60
SGBDNL - ERP	○	99.07	64.97	95.99	63.06	33	71
SGBDNL - MailQueue	○	99.60	63.60	97.17	72.56	26	61
SGBDNL - Solr	○	89.90	65.33	78.08	65.38	20	60
SGBDNL - Loyalty	○	92.85	59.84	91.57	68.23	14	56
SGBDNL - Reporting	○	96.63	73.69	99.07	75.21	12	35
SGBDNL - FeatureAdvantage	○	99.99	61.05	94.18	69.99	11	60
RedMine	□	80.87	64.15	85.12	51.10	0	18
SGBDNL - EBPSQL	○	34.42	47.90	43.96	72.43	0	77
SGBDNL - PhpSgdOnline	○	43.45	56.01	51.67	79.64	0	67

Cloud Migration Context – Survey Answers

Key Insights	Description	Comments
Cloud Technologies Maturity	What is the average skill on Cloud technologies & practices within your development team?	**High Maturity** - 13 out of 14 equal "Experts" - 1 out of 14 equals "None"
Evolution Model	What is your evolution model & feedback loo implementation	**CI/CD Environment** - 13 out of 14 are set as Continuous Delivery - 1 out of 14 is set as Waterfall
Multi-Tenancy	Is this application multi-tenant	**Multi-tenancy Practice** - 10/14: multi-tenant with dedicated DB - 3/14: full multi-tenant - 1/14: single-tenant
Database Compliance	What is the application database provider	**Standard Databases** - 11/14: Standard but not available in IaaS - 3/14: Standard and supported in PaaS

Cloud Boosters & Blockers – WCF - EBusiness Platform Application

60.8	54.1	67.6	8.5	17.6	199
CloudReady	CloudReady Survey	CloudReady Scan	Boosters	Blockers	Roadblocks

Cloud Requirement	Impact	Criticality	Contribution	Roadblocks
Persistent Files : Perform File Manipulation	CFA	Medium	- 4.41 %	2
Code Execution : Using COM Components	CF	High	- 4.13 %	18
Data Encryption Key : Using Crypto API	CFA	Low	- 2.20 %	1
Execution Environment : Using hardcoded network IP address (IPV4, IPV6)	C	Low	- 2.20 %	57
Security & User Authentication : Using of unsecure network protocols (HTTP, FTP)	C	Low	- 2.20 %	118
Sensitive Data Storage Protection : Using RDBMS Access	CFA	Low	- 2.20 %	2
Application Settings Configuration : Using other configuration files than Web configuration	C	Low	- 0.28 %	1
Application Settings Configuration : Using ConfigurationManager		Low	+4.26 %	0
Azure Registry Settings : Using a Cloud-based storage		Low	+4.26 %	0

Cloud Boosters & Blockers – WCF - EBusiness Platform application

Description	Comments
The team in SGBD BNL is one **of the most advance and expert BU** in Saint-Gobain's group regarding Cloud expertise	The team is used to Cloud development and will be able to identify quickly the blockers for a PaaS migration
The application is develop **in C# .NET with the WCF component**	There is some **official documentation about this migration** but it has some important pitfalls. The team was clear on this and this service is being rewritten along the time. So we decided **not to go for a PaaS migration** for the moment. The **component will be transform in serverless** later.
Blockers are **File manipulation, usage of COM components, unsecured URL (HTTP), and configuration**	The **main blocker is the COM components registration**, this will be not so easy to overcome as it is the core of the WCF. The file manipulation is replaceable, even easier for the URL and the configuration files.

Cloud Boosters & Blockers – SGBDBNL – API Application

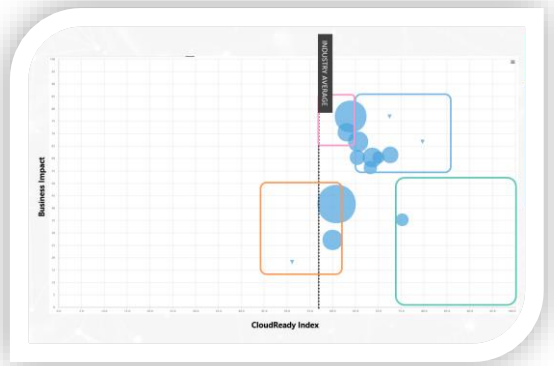
63.9	54.1	73.7	12.8	11.0	129
CloudReady	CloudReady Survey	CloudReady Scan	Boosters	Blockers	Roadblocks

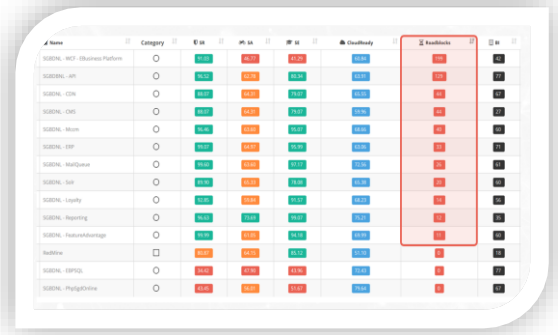
Cloud Requirement	Impact	Criticality	Contribution	Roadblocks
Code Execution : Using COM Components	CF	High	- 4.13 %	18
Execution Environment : Using hardcoded network IP address (IPV4, IPV6)	C	Low	- 2.20 %	27
Security & User Authentication : Using of unsecure network protocols (HTTP, FTP)	C	Low	- 2.20 %	81
Sensitive Data Storage Protection : Using RDBMS Access	CFA	Low	- 2.20 %	2
Application Settings Configuration : Using other configuration files than Web configuration	C	Low	- 0.28 %	1
Application Logs : Correct usage of Logging		Low	+4.26 %	0
Application Settings Configuration : Using ConfigurationManager		Low	+4.26 %	0
Azure Registry Settings : Using a Cloud-based storage		Low	+4.26 %	0

Cloud Boosters & Blockers – SGBDBNL - API

Description	Comments
The context is different on this application because it is already an API application with a modern design, align with technology choices and easy to maintain	The team is used to Cloud development and will be able to identify and modify quickly the blockers for a PaaS migration
The low point is the usage of a **Standard Database provider which is SQL Server.**	The survey identify this as roadblock (IaaS only) but in fact this is **not a blocker to move to a PaaS service** the application does not rely on a Database it just uses connectors to other DB services.
The application is **well ranked by the code source scan**, it is the survey that takes it down, but beside this the application is **very resilient and well written.**	This application is a good candidate for a cloud migration. We decided to **go for a PaaS migration as this app will be central to others in the cloud.**
Blockers are **File manipulation, unsecured URL (HTTP), and configuration**	The COM components registration, and others **blockers are easy to overcome, because they only apply to the Unit testing part of the application**. The file manipulation is easily replaceable, the configuration is straightforward on this.

Cloud Migration Strategy | **INSIGHTS**





- Define the extent of the Cloud migration
- Identify quick wins and Core cloud apps (the ones that others will rely on)
- Don't stop to the results, drill down with the team to understand what really are the blockers
  - Maybe it is just for dev environments or specifics scenarios
- Determine the coding effort vs the maintenance effort
- Start with quicks win to reassure everyone
- Give insights to your customers
- Go the extra mile and propose working cloud architecture tailored to their applications

www.casthighlight.com

Thank you