



BENCHMARKING SOBRE EL RIESGO EN APLICACIONES CRÍTICAS

¿EL MERCADO SIEMPRE MANDA?

¿Cuál es el valor de la medición?

- Una buena medición tiene que ser repetible, consistente, basada en estándares, precisa y por lo tanto creíble.
- La medición tiene que soportar la toma de decisiones o acciones - que pueden ser del tipo:
 - Aumentar la supervisión
 - Verificación utilizando otras fuentes
 - Investigar causa raíz
 - Cambiar procesos
 - Cambiar prioridades
 - Establecer acciones de remediación
 - Disponer de información objetiva
 - Gestión de Deuda Técnica
 - Etc.

La Analítica de Software ofrece valor a través de múltiples dimensiones

El análisis de software es el uso inteligente de la información sobre los aplicativos para mejorar las decisiones de inversión TI, el rendimiento operacional y resultado entregado a los clientes.

PREVENIR RIEGOS EN EL NEGOCIO

- Identificar los riesgos que afectan a los procesos de negocio, la satisfacción del cliente y e impactan en la marca de empresa
- Mejorar la reputación como líder basado en datos analíticos

“Organizaciones que se basan en datos analíticos superan **2.2 veces a su industria**, y reconocen que la analítica verdaderamente le ayuda a ser más competitivos

- IBM AND MIT SLOAN



AUMENTAR INGRESOS

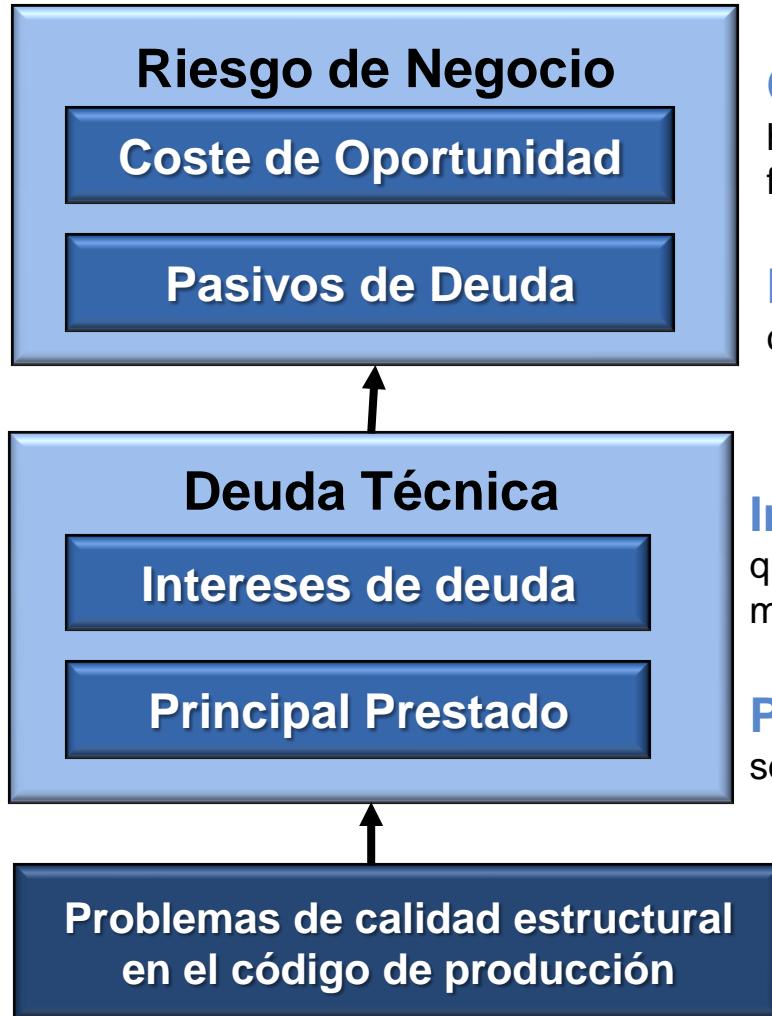
- Mejorar la agilidad de la organización
- Mejorar el Time-to-market (Coste de Oportunidad)
- Mejorar la productividad del AM

REDUCIR GASTOS EN AM

- Monitorizar Deuda Técnica
- Obtener más por lo mismo
- Obtener más con menos (reducir recursos infrautilizados, proyectos con bajo rendimiento , aplicaciones infratratilizadas, etc.)
- Negociar mejor con CFO el presupuesto

Metáfora de la Deuda Técnica

Deuda Técnica — Coste futuro de los defectos presentes en el código,
es un componente del coste de propiedad.



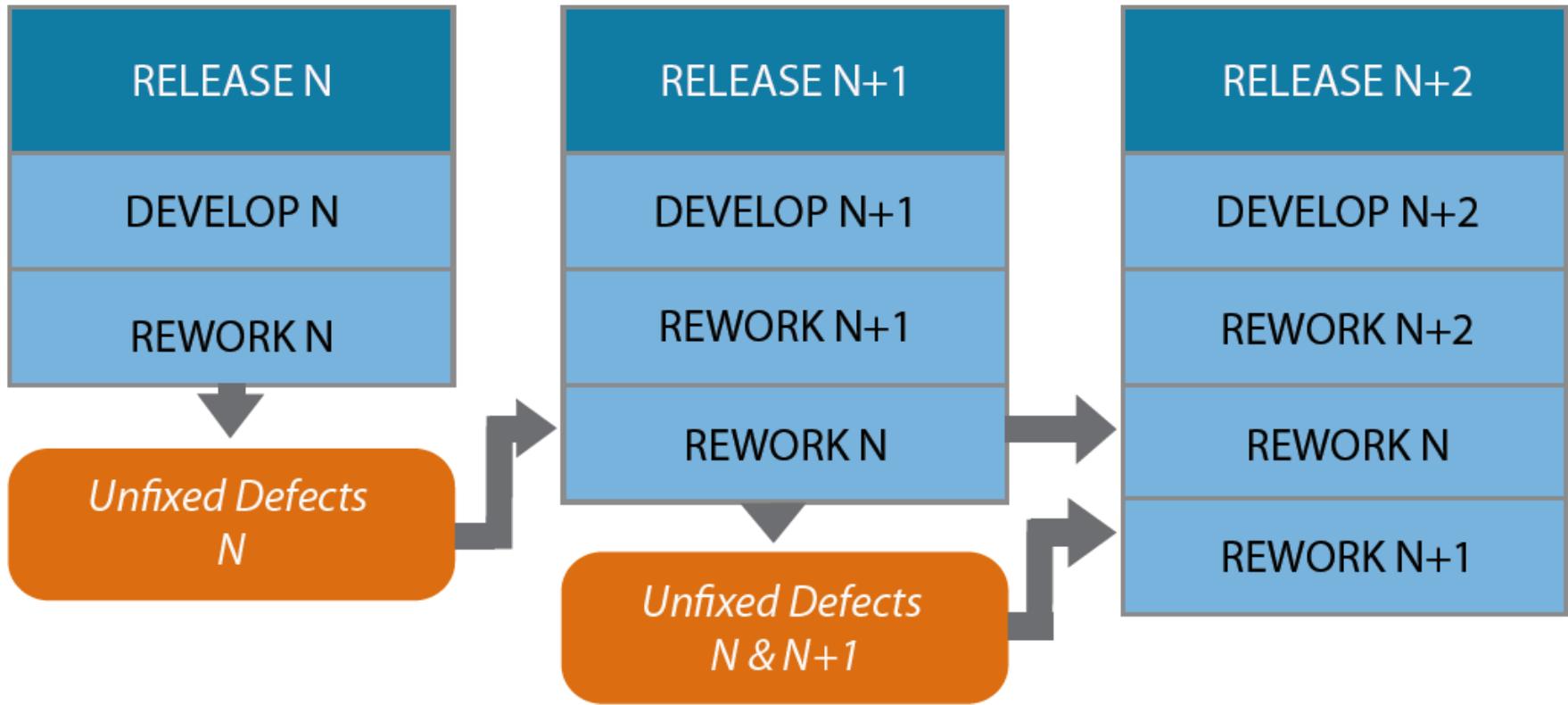
Coste de Oportunidad Beneficios que se podían haber obtenido si los recursos puesto en una nueva funcionalidad en lugar de reducir deuda técnica.

Responsabilidad Costes para el negocio debidos a caídas, datos corruptos, riesgos de hacking, etc.

Interés Costes TI atribuidos a los defectos (violaciones) que producen Deuda Técnica incluyendo altos costes de mantenimiento y uso de recursos técnicos, humanos, etc.

Principal Coste de eliminar problemas que deben de ser solucionados en el código entregado.

Deuda Técnica – Carry-forward rework



Liderando la industria de IT Software Analytics

CAST con los

ESTÁNDARES

La adopción de los puntos de función, robustez, seguridad y eficiencia expande drásticamente la eficacia del software analytics .



CAST debería incluirse

BASE DE SLA'S

para la calidad del software en los contratos ADM de proveedores.



CAST es

DE FACTO

El Estándar para medir Calidad de Software

Gartner

CAST es

LÍDER

en el mercado TI.



CAST lidera la

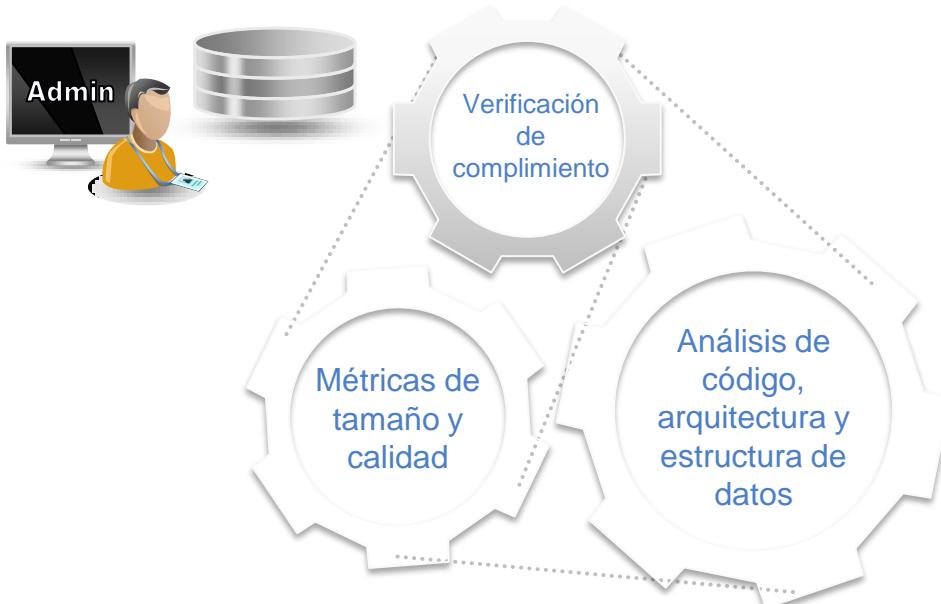
TECNOLOGÍA

de esta clase



CAST Application Intelligence Platform

The CAST Application Intelligence Center



Visibilidad para toda la organización IT:

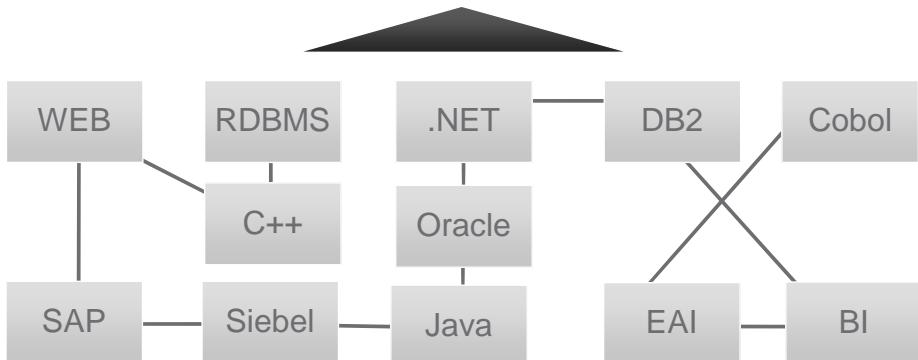
Tamaño Técnico

Tamaño Funcional

Calidad Estructural

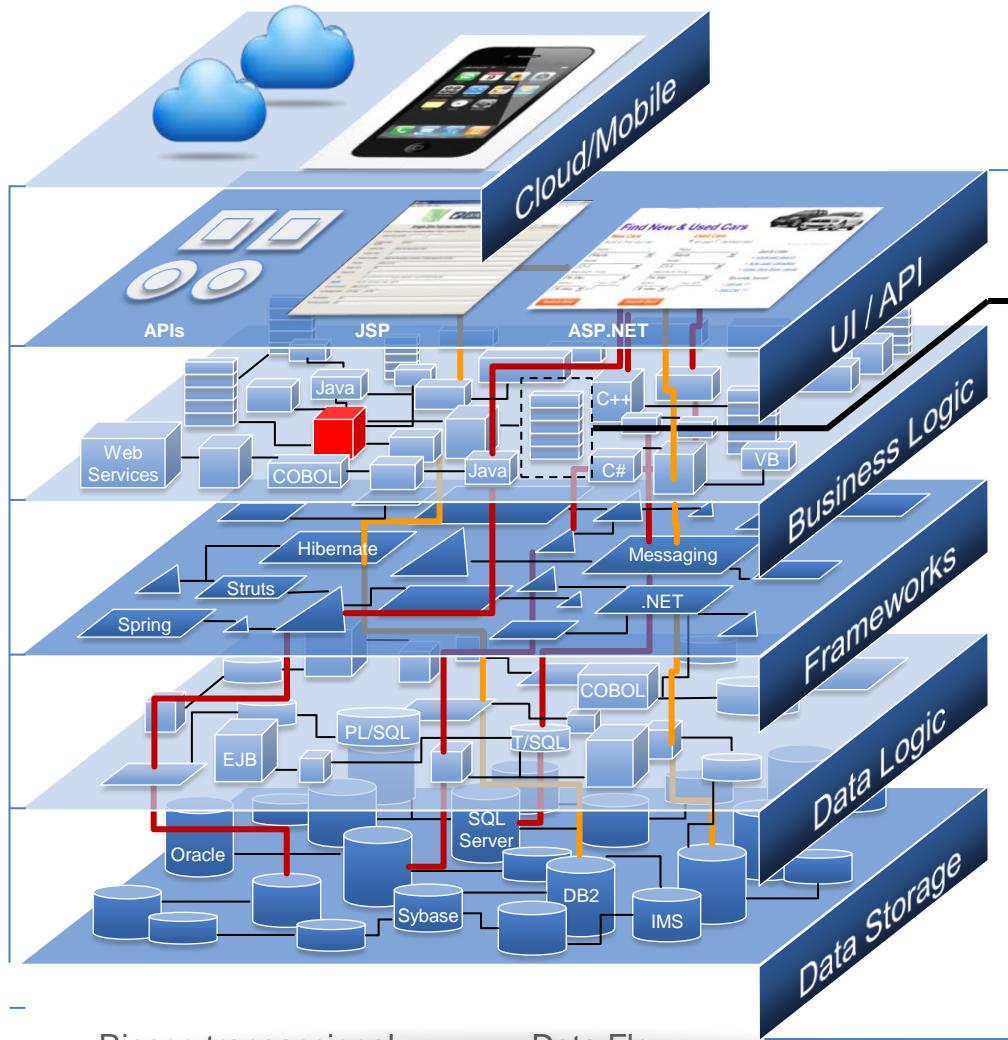
Deuda Técnica

Benchmarking
appmarq



La visión entre tecnologías y entre capas

Cumplimiento Arquitectura



Nivel de programa

- Reglas de estilo codificación
- Complejidad de expresiones
- Documentación de código
- Diseño de programas o clases
- Estándares de codificación

1

Nivel de Módulo

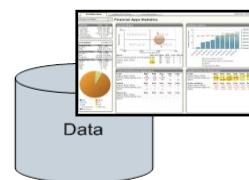
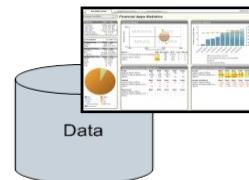
- Arquitectura Inter-tecnología
- Dependencias entre capas
- Complejidad y cohesión
- Diseño y estructura
- Invocación entre programas
- Vulnerabilidades de seguridad

2

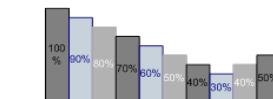
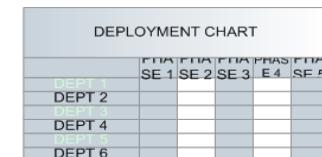
Nivel de Sistema

- Calidad de integración
- Cumplimiento arquitectura
- Simulación de riesgo de propagación
- Seguridad de aplicación
- Pruebas de resistencia
- Integridad transaccional
- Medición de Puntos Función
- Estimaciones de esfuerzo
- Control de acceso a datos
- Versionados de SDK
- Calibración de tecnologías

3

Automated Structural Quality Measurement**CAST AIP Structural Quality Measurements****Data Extraction**

Automated extraction and upload to Appmarq repository

Appmarq Benchmarking Repository**Benchmarking Analyses**

- Appmarq es un repositorio de datos de calidad estructural de aplicaciones de software personalizadas en el negocio de TI.
- Los datos se recogen a través del análisis automatizado con CAST Application Intelligence Platform (AIP), que realiza un análisis de la calidad estructural tanto a nivel de código como aplicación.
- Las métricas obtenidas se introducen en el repositorio del Appmarq. Todos los datos se almacenan de forma totalmente anónima y normalizada antes de su evaluación comparativa.
- Actualmente, es la mayor muestra de aplicaciones medidas mediante estándares y teniendo en cuenta su calidad estructural a través de sus diferentes tecnologías que hay disponible en el mercado.



1. Métricas de Calidad Estructural: Appmarq contiene métricas de calidad estructural de aplicaciones, no de coste, esfuerzo, etc.



2. Generación de Datos automatizado: Los datos de Appmarq se generan a través de un análisis objetivo automatizado de aplicaciones en lugar de encuestas subjetivas.



3. Aplicaciones del Mundo Real: Los datos de Appmarq pertenecen a aplicaciones que actualmente están en operación.



4. Conjunto de Datos Diverso: Appmarq contiene datos de calidad estructural de un amplio rango de sectores industriales, tecnologías y geografías.

Diferentes Perspectivas de Benchmarking

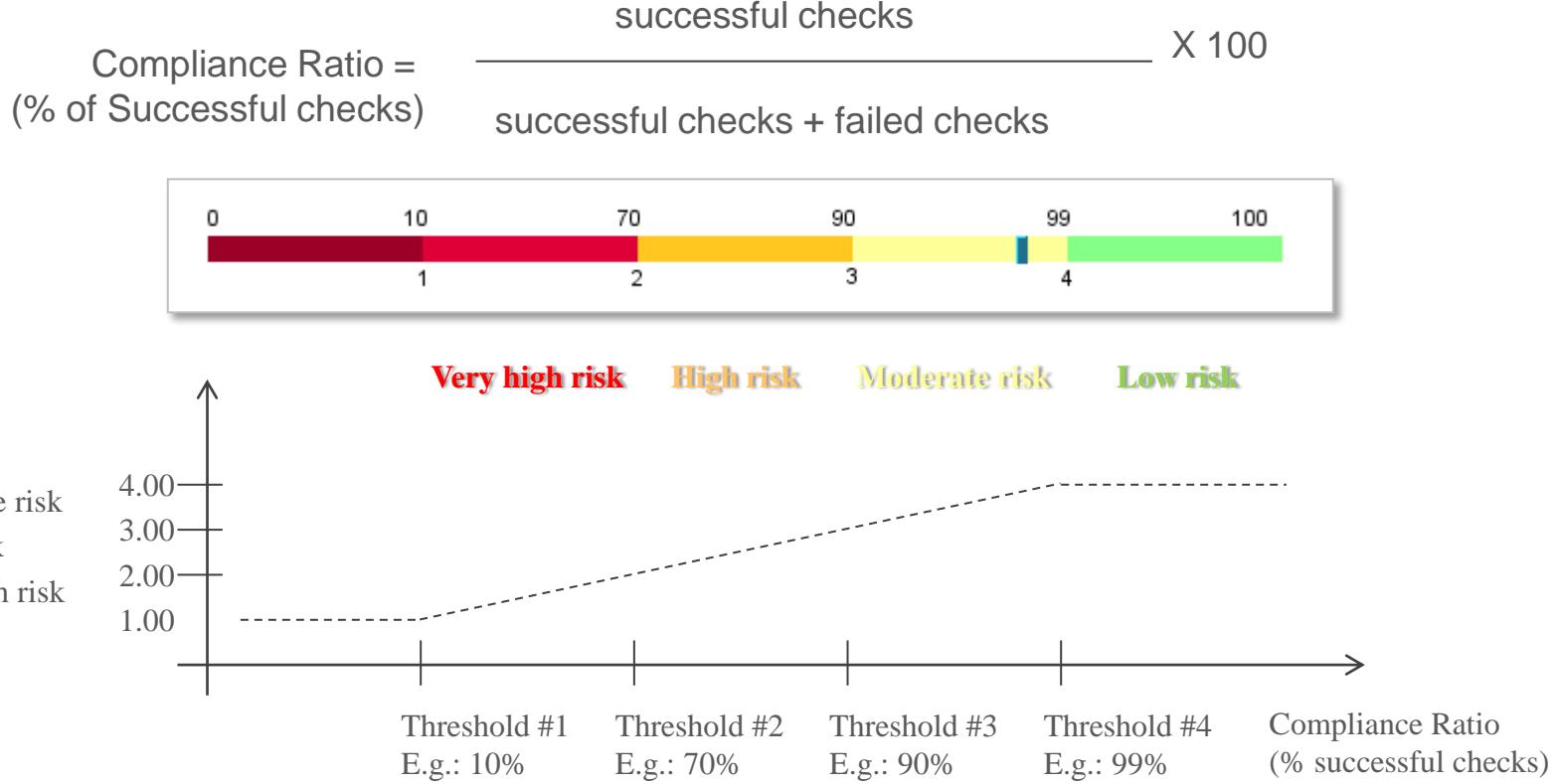
- Tecnología
- Factores Demográficos
 - Sector de la industria
 - Origen
 - In-House
 - Outsourced
 - Shore
 - Onshore
 - Offshore
 - Nivel de madurez CMMI
 - Metodología de Desarrollo
 - Número de Usuarios

Factores de Salud

Factor	Descripción	Valor para la Empresa
Rendimiento	Nivel de riesgo sobre el rendimiento actual y futuro de la aplicación en función de su concepción y su arquitectura.	Un mejor índice de rendimiento reduce el riesgo de que una aplicación consuma demasiados recursos y mejora su escalabilidad.
Robustez	<ul style="list-style-type: none">■ Nivel de nivel de riesgo o de probabilidad de introducir errores o fallos en la aplicación durante modificaciones (estabilidad).■ Nivel de esfuerzo necesario para realizar el <i>Testing</i> de la aplicación.	<p>Una mayor robustez permite</p> <ul style="list-style-type: none">■ Evitar fallos y errores.■ Mejorar la satisfacción de los usuarios y la imagen de la organización de TI.
Seguridad	Nivel de riesgo de fallos potenciales de seguridad en una aplicación.	Mejores notas en seguridad disminuyen los riesgos de daño a la aplicación.
Transferencia	Nivel de esfuerzo necesario para transferir el conocimiento sobre la aplicación.	Contribuye a: <ul style="list-style-type: none">■ Evitar la dependencia respecto a recursos internos o proveedores de servicios.■ Mejorar la productividad de los equipos.■ Facilitar la transferencia y entrada de externos y de equipos internos o viceversa.
Cambio	Nivel de esfuerzo para implementar una modificación (evolución o corrección) dentro de una aplicación.	Una mejor capacidad de cambio contribuye a: <ul style="list-style-type: none">■ Mejorar la facilidad y la rapidez del mantenimiento.■ Mejorar la previsión de entregables.■ Acelerar la salida al mercado.
Mantenimiento (SEI)	Coste y dificultad / facilidad de mantener una aplicación en el futuro.	Un mejor índice de mantenimiento disminuye los costes de mantenimiento de las aplicaciones.

El por qué de la nota de 1 a 4

- Calculado por regla, componentes, módulos y aplicaciones
- Por debajo de 3, los riesgos y costes aumentan



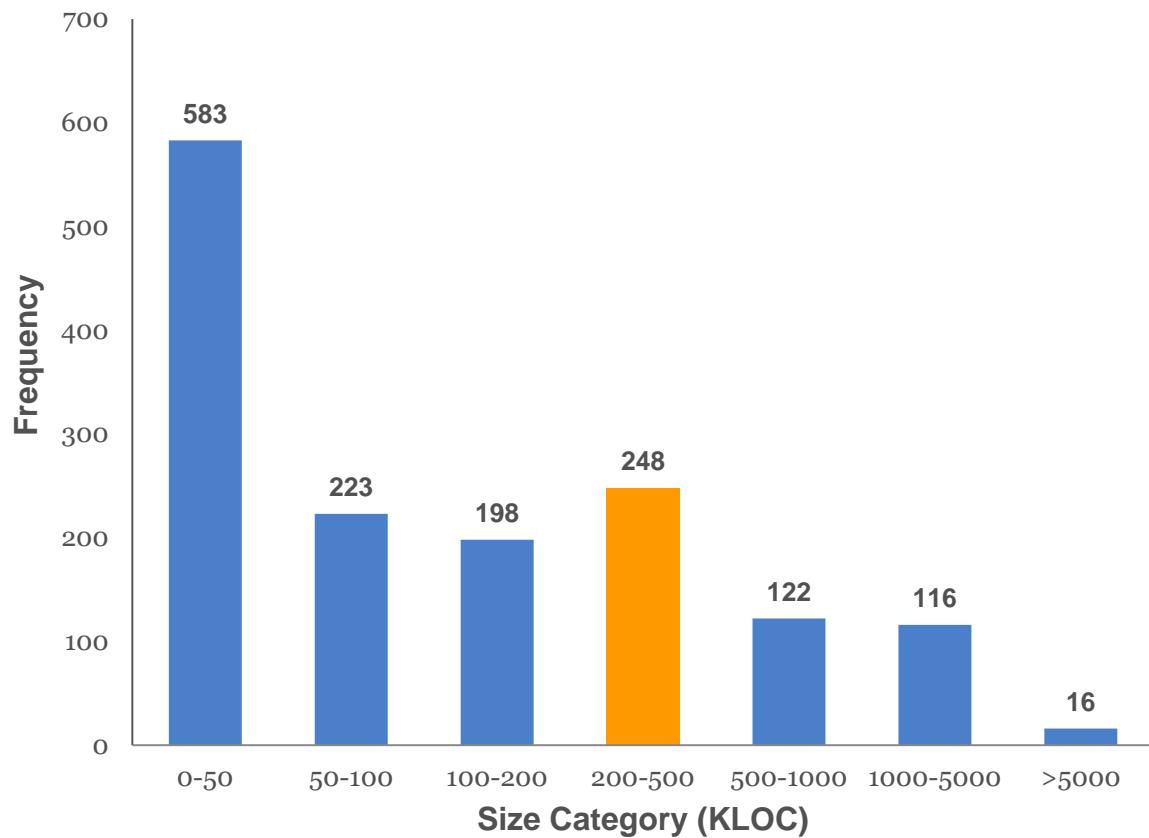
Los datos de este informe de evaluación comparativa se extrae de una base de datos compuesta por **1744 aplicaciones**, lo que representa **1.861 M líneas de código**.

Estos datos han sido recogidos durante un período de 7 años y ha sido compartidos por **291 organizaciones** para el análisis estático de sus características de calidad estructural.

Estas organizaciones representan a **14 segmentos de la industria**: Servicios Financieros, Seguros, Telecomunicaciones, Manufactura, Gobierno, Servicios Públicos, Retail, IT & Business Consulting y otros.

Las organizaciones se encuentran principalmente en Norteamérica, Europa e India.

Distribución de Aplicaciones en base a su Categoría por Tamaño



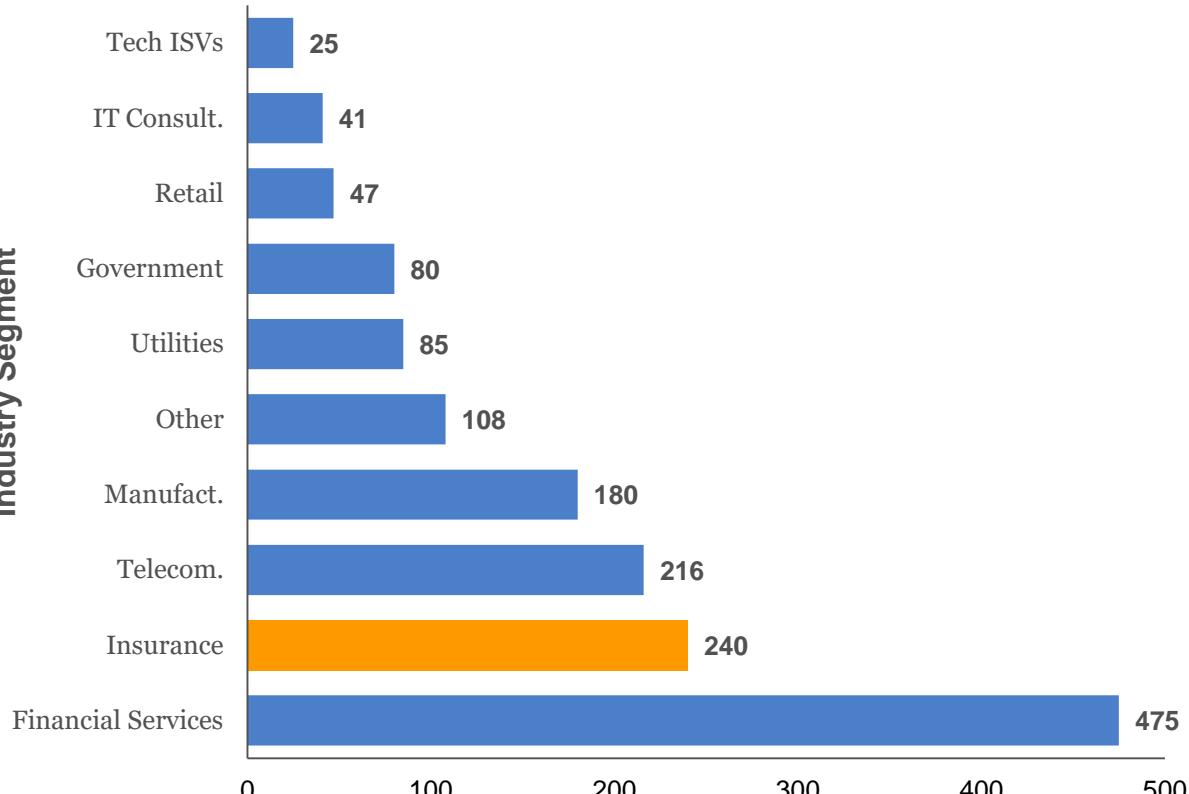
Key Figures

- Data ranges from 1 to 52,773 KLOC
- Average application size in Appmarq is about 424 KLOC
- Applications are widely distributed across size categories increasing the representativeness of the Appmarq sample
- Category2 application represents **352 thousand lines of code** finding itself below average in size for this dataset

	0-50	50-100	100-200	200-500	500-1000	1000-5000	>5000
Total Apps	583	223	198	248	122	116	16
# Companies	106	84	85	102	72	63	12

Industry Segment Distribution across Applications

Industry Segment

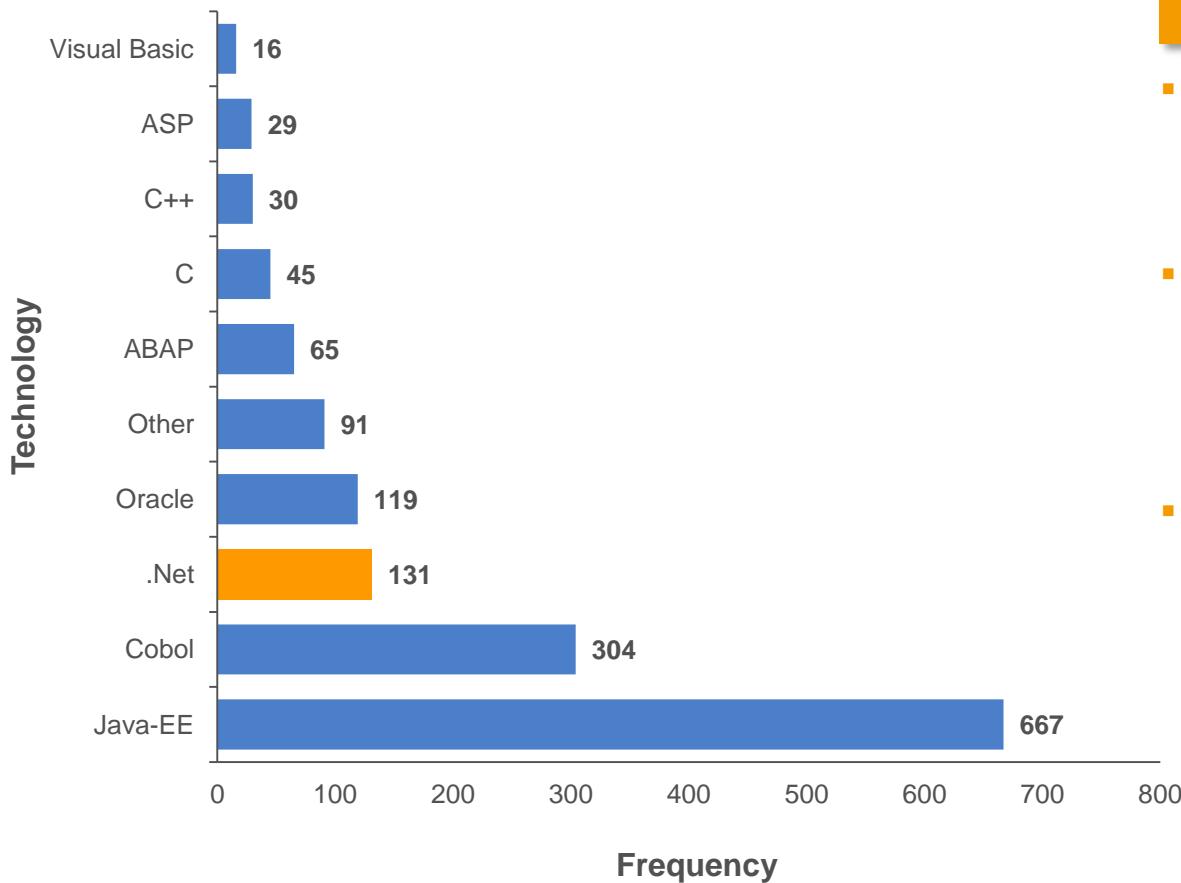


Key Figures

- Data are drawn from 14 main industry segments, which represent the primary business activity of the organization for which the application was developed
- Category2 application has been classified under the **Insurance** industry segment

	Financial Services	Insurance	Telecom.	Manufact.	Other	Utilities	Governm	Retail	IT Consult.	Tech ISVs
Total Apps	475	240	216	180	108	85	80	47	41	25
# Companies	49	31	20	26	32	15	16	11	16	8

Technology Distribution across Applications

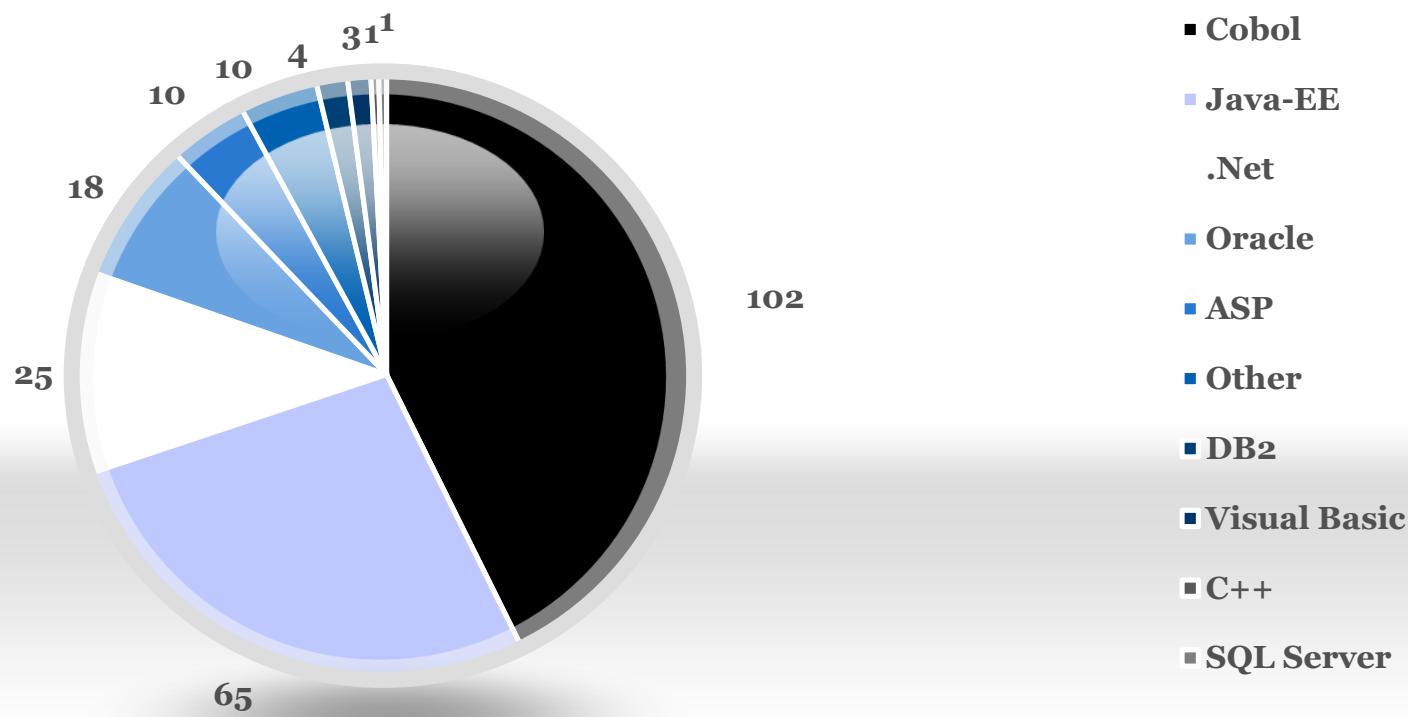


Key Figures

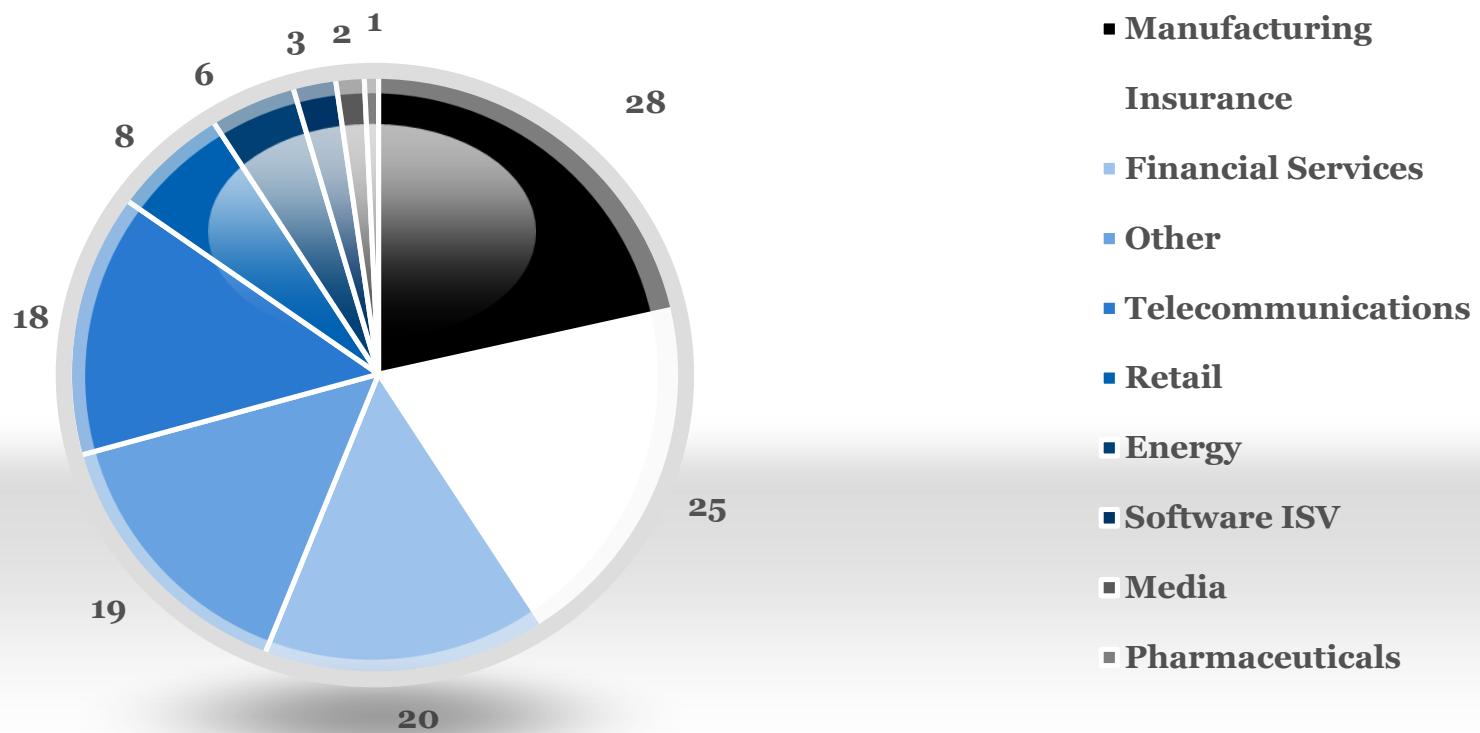
- Applications are categorized according to the most dominant technology
- The largest concentration of applications is in Java-EE, representing 45% of the total sample
- Category2 application is primarily made up of .Net technology

	Java-EE	Cobol	.Net	Oracle	Other	ABAP	C	C++	ASP	Visual Basic
Total Apps	667	304	131	119	91	65	45	30	29	16
# Companies	135	42	57	45	49	25	17	16	10	10

Insurance Break-Down Across Technology



.Net Break-Down Across Industry Segment



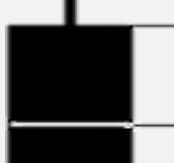
Interpreting Box and Whisker Plots

In this report, most of the data is represented in box and whisker plots in order to see the distribution of results

Maximum: Highest data point regardless of whether it is an outlier

 **Outlier:** a data point greater than 2 standard deviations from the mean

 **Upper whisker:** Highest data point that is not an outlier

 **Upper Quartile:** 75% of the data is below this point

 **Median:** 50% of the data is greater than this value (mid-point of the distribution)

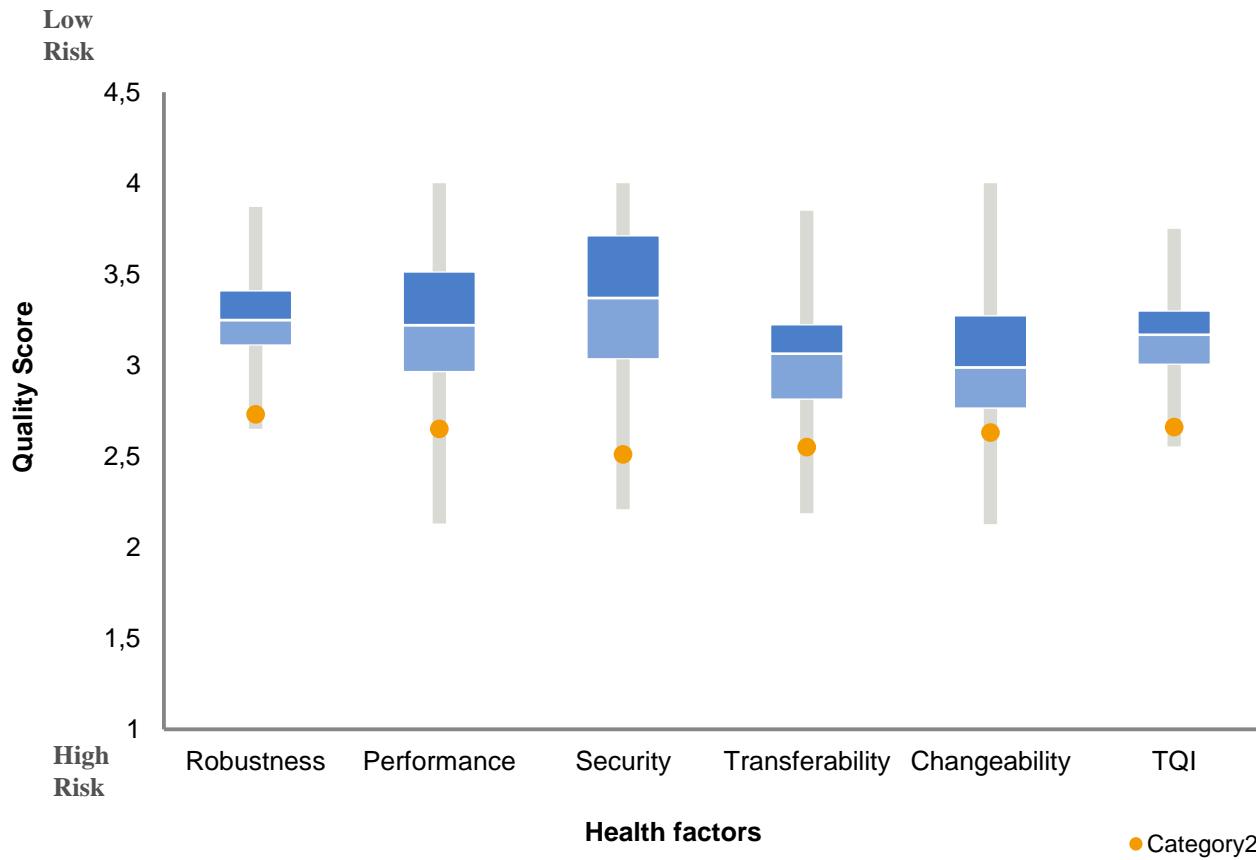
 **Lower Quartile:** 25% of the data is below this point

 **Lower whisker:** Lowest data point that is not an outlier

Minimum: Lowest data point regardless of whether it is an outlier

Each boxplot in this report is followed by the descriptive statistics that characterize each of its major boundaries. **Outliers** may not be shown on plots, but **cause the maximum or minimum values to differ from the upper or lower whisker**

Benchmark Across Industry – Insurance

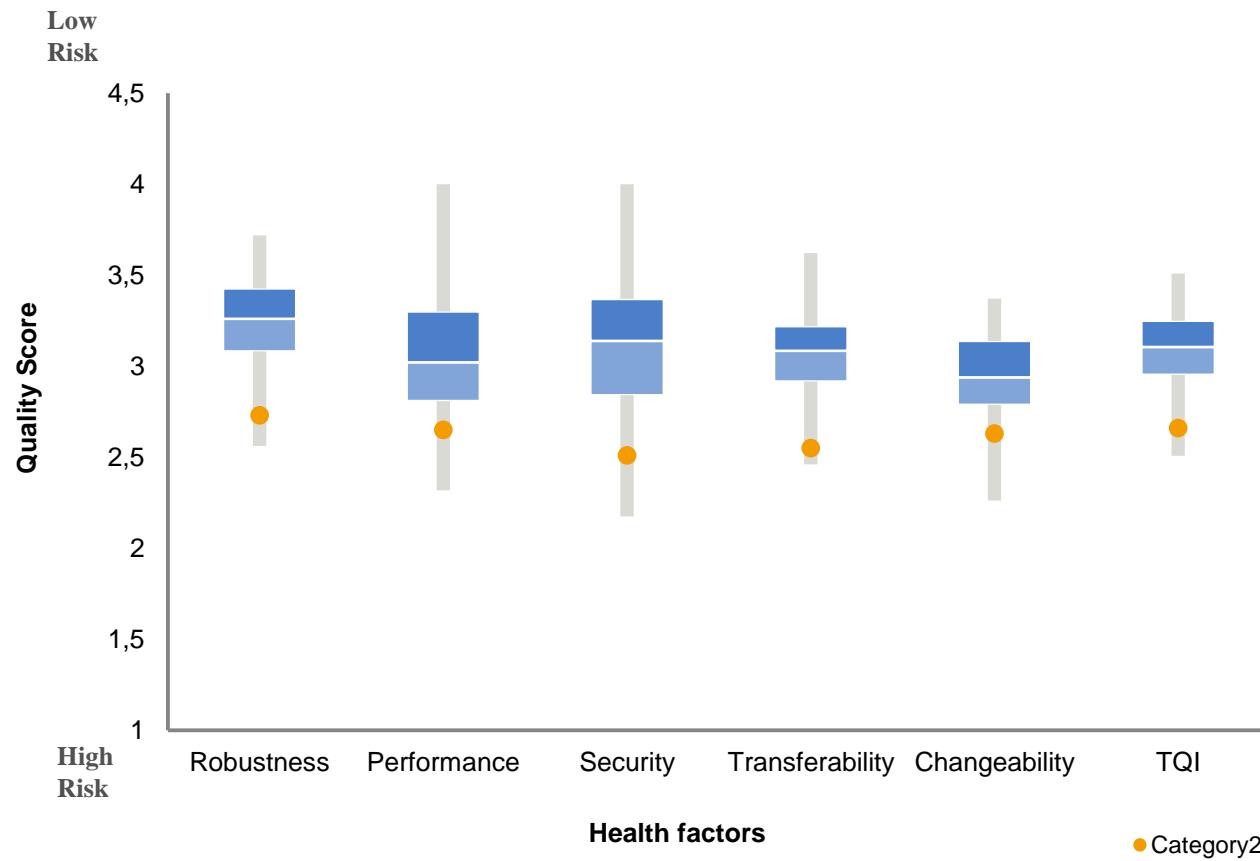


Statistics for Benchmark Across Industry – Insurance

	Robustness	Performance	Security	Transferability	Changeability	TQI
Minimum	2.38	1.99	2.21	2.01	2.13	2.49
Lower Whisker	0.46	0.83	0.82	0.62	0.63	0.45
25 th Percentile	3.11	2.96	3.03	2.81	2.76	3.00
Median	3.25	3.22	3.37	3.06	2.99	3.17
75 th Percentile	3.41	3.52	3.71	3.23	3.27	3.30
Upper Whisker	0.46	0.48	0.29	0.62	0.73	0.45
Maximum	4.00	4.00	4.00	4.00	4.00	4.00
Category2	2.73	2.65	2.51	2.55	2.63	2.66

- Application benchmarked against 239 applications, drawn from 31 companies, representing 107,989 thousand lines of code in the Insurance segment
- Category2 application score below average on all quality characteristics compared to other applications in Insurance industry. Category2 application has the lowest score for Security, Performance, Robustness
- This application proves at risk of unauthorized intrusions, less efficient in terms of performance in its industry, less stable and robust in its industry

Benchmark Across Technology – .Net

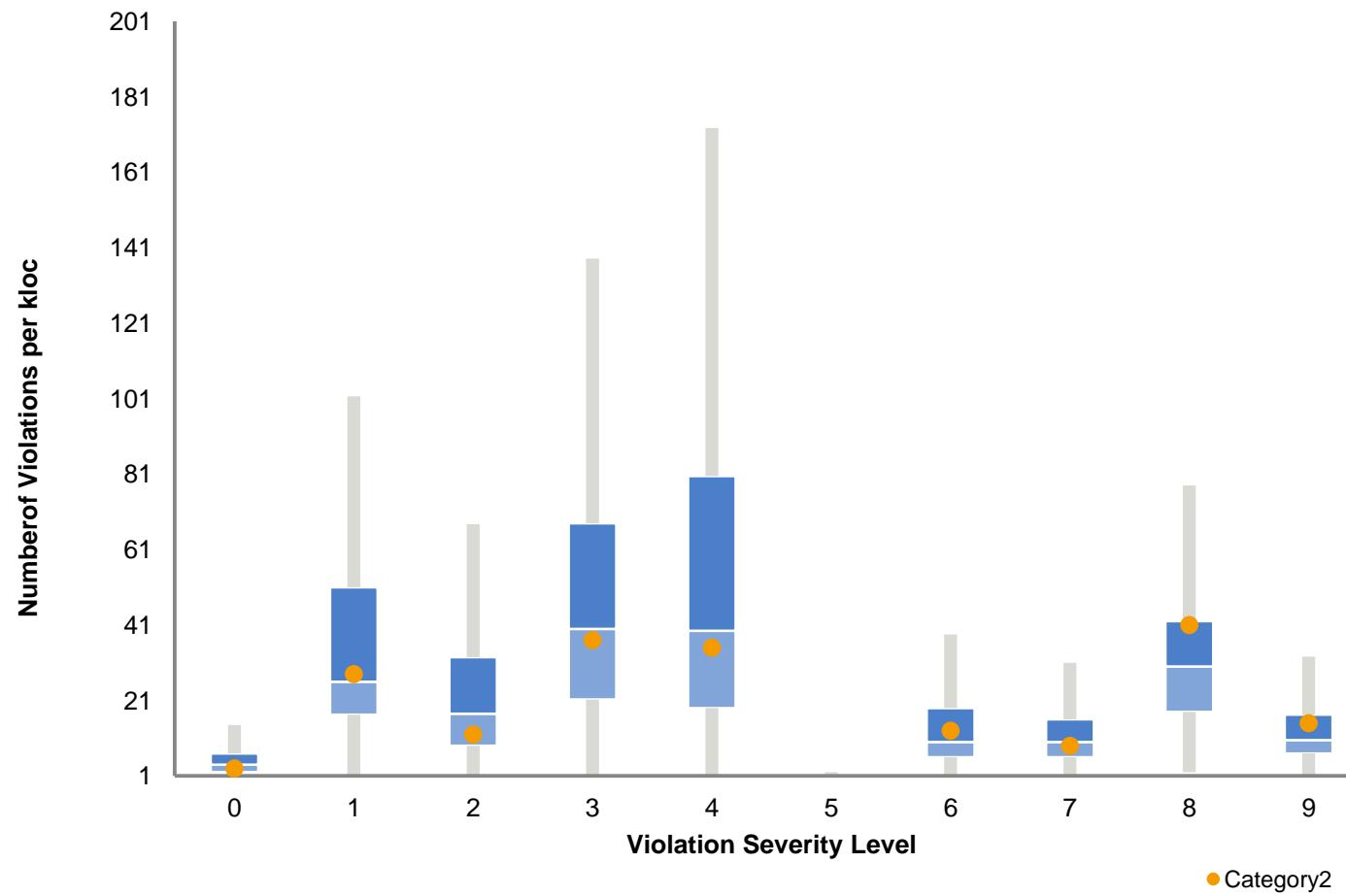


Statistics for Benchmark Across Technology – .Net

	Robustness	Performance	Security	Transferability	Changeability	TQI
Minimum	2.17	2.32	2.17	2.23	2.17	2.24
Lower Whisker	0.52	0.49	0.67	0.46	0.53	0.45
25 th Percentile	3.08	2.81	2.84	2.92	2.79	2.95
Median	3.26	3.02	3.14	3.08	2.94	3.11
75 th Percentile	3.43	3.30	3.37	3.22	3.14	3.25
Upper Whisker	0.29	0.70	0.63	0.40	0.23	0.26
Maximum	3.72	4.00	4.00	3.62	3.37	3.51
Category2	2.73	2.65	2.51	2.55	2.63	2.66

- Application benchmarked against 129 applications, drawn from 57 companies, representing 35,460 thousand lines of code of .Net technology
- Category2 application score below average on all quality characteristics compared to other applications made up of .Net technology. Category2 application has the lowest score for Security, Transferability, Robustness
- This application proves at risk of unauthorized intrusions, more difficult for the development team to understand and transfer on to others, less stable and robust compared to others in the same technology

Benchmark Across Violation Density per Severity Level in .Net Applications



Statistics on Violation Density Benchmark

	0	1	2	3	4	5	6	7	8	9
Minimum	1	1	1	1	1	1	1	1	2	1
Lower Whisker	1	16	8	20	18	0	5	5	16	6
25 th Percentile	2	17	9	21	19	1	6	6	18	7
Median	4	26	18	40	40	1	10	10	30	11
75 th Percentile	7	51	33	68	81	2	19	16	42	17
Upper Whisker	8	51	35	70	92	1	20	15	36	15
Maximum	465	412	270	357	239	2	401	46	203	490
Category2	3	28	12	37	35	0	13	9	41	15

- The number of violations per thousand lines of code has been calculated for each .Net application in Appmarq. Each rule that has been violated holds a severity level from 0 (low) to 9 (high) which represents the likely damage it will cause if the violation is encountered in operation or maintenance
- Category2 application has a fairly **high violation density across critical violations** (severity levels 7, 8, and 9)

Technical Debt Formula

Parameters used for calculating estimates of Technical Debt—Principal

Violation Severity Level	% Violations to be fixed	Hours to fix	Cost / Hour
High (7-9)	100%	2.56	\$75
Medium (4-6)	50%	0.97	\$75
Low (0-3)	0%	-	-

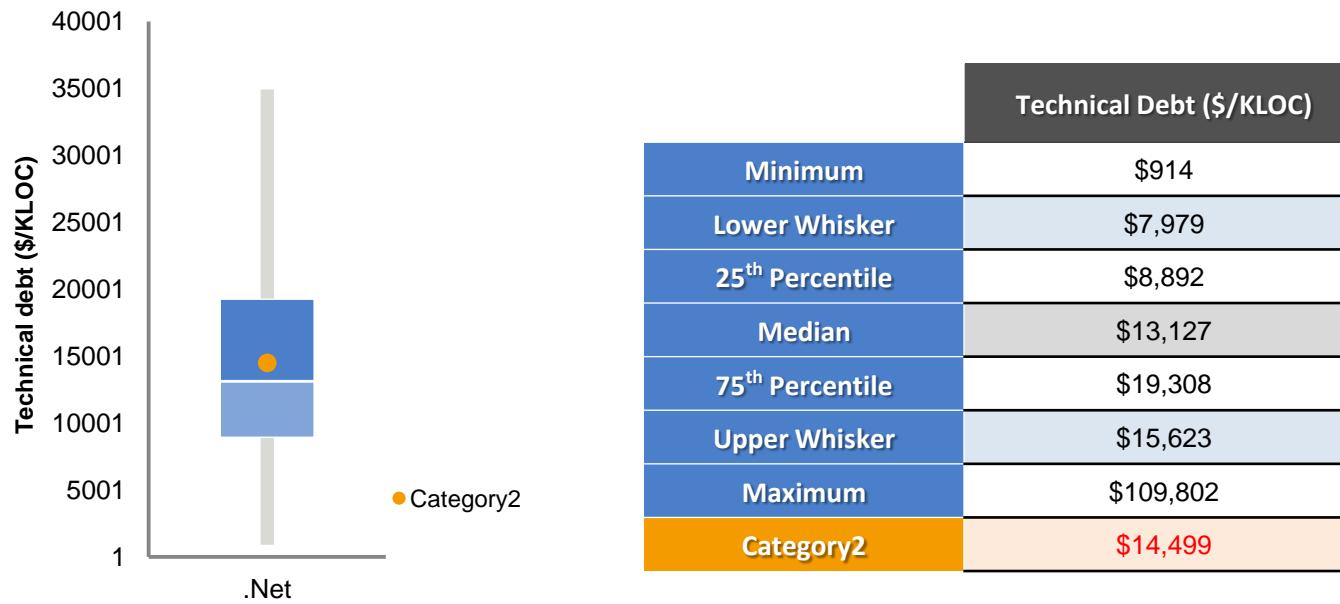
Estimated Technical Debt—Principal =

(Σ high severity violations) X (% to be fixed) X (average hours to fix) X (\$s per hour) +
(Σ medium severity violations) X (% to be fixed) X (average hours to fix) X (\$s per hour) +
(Σ low severity violations) X (% to be fixed) X (average hours to fix) X (\$s per hour)

The formula provides a very **conservative estimate** of Technical Debt—Principal

IT organizations can get more accurate estimates by adjusting the parameters in the formula based on their experience

Benchmark of Technical Debt Principal for .Net Applications



- The calculation of the Technical Debt Principal has been normalized and benchmarked across the thousand lines of code within the .Net technology group
- Category2 application has a total Technical Debt of \$5,103,789, which represents **\$14,499 per thousands lines of code**
- Compared to other .Net applications, this amount is **above average**

Conclusiones I

- Los efectos más fuertes entre todas las variables demográficas medidas fue la madurez de los procesos:
 - Organizaciones CMMI Nivel 1 producen aplicaciones con resultados sustancialmente más bajos en todos los factores de salud que las aplicaciones desarrolladas en CMMI Nivel 2 o Nivel 3 organizaciones. Mientras que la madurez del proceso representó el 11% o más de la variación entre todas las puntuaciones de los factores de salud, especialmente en robustez y seguridad.
- El impacto de la metodología de desarrollo no es tan grande como el de la madurez de proceso.
 - No obstante, en función de la metodología se ven afectados los resultados de los factores de la salud y representa entre un 5% y el 15% de variación.
 - A través de todos los factores de salud, un híbrido de metodología mezcla de Agile y Waterfall producen puntuaciones que si se aplican cada una de forma ortodoxa.

Conclusiones II

- El número de usuarios atendidos por una aplicación tiene un efecto similar al de las metodologías de desarrollo
 - Representa entre el 4% y el 14% de la variación entre las puntuaciones de los factores de la salud. Las aplicaciones que sirven a más de 5000 usuarios tenían puntuaciones más altas en todos los factores de salud.
- El sector de la industria en la que se ha desarrollado una aplicación tiene un pequeño efecto sobre su Seguridad, Cambiabilidad y Transferencia
- La decisión de desarrollar aplicaciones in-house frente a la externalización no tuvo efecto sobre las puntuaciones de calidad estructural .
 - La elección de desarrollar aplicaciones onshore versus offshore tiene efectos mínimos sobre Cambiabilidad y Robustez.

- The CRASH Report 2014-2015 (CAST Research on Application Software Health)
 - Enlace de Descarga.
<http://www.castsoftware.com/research-labs/crash-reports>



The CRASH Report 2014-2015
(CAST Research on Application Software Health)

The Global State of Structural Quality in IT Applications



- Demo CAST AIP Application Analytics Dashboard
 - Enlace: <http://demo-eu.castsoftware.com/AAD/portal/index.html>
 - Usuario: CIO
 - Password: CAST

