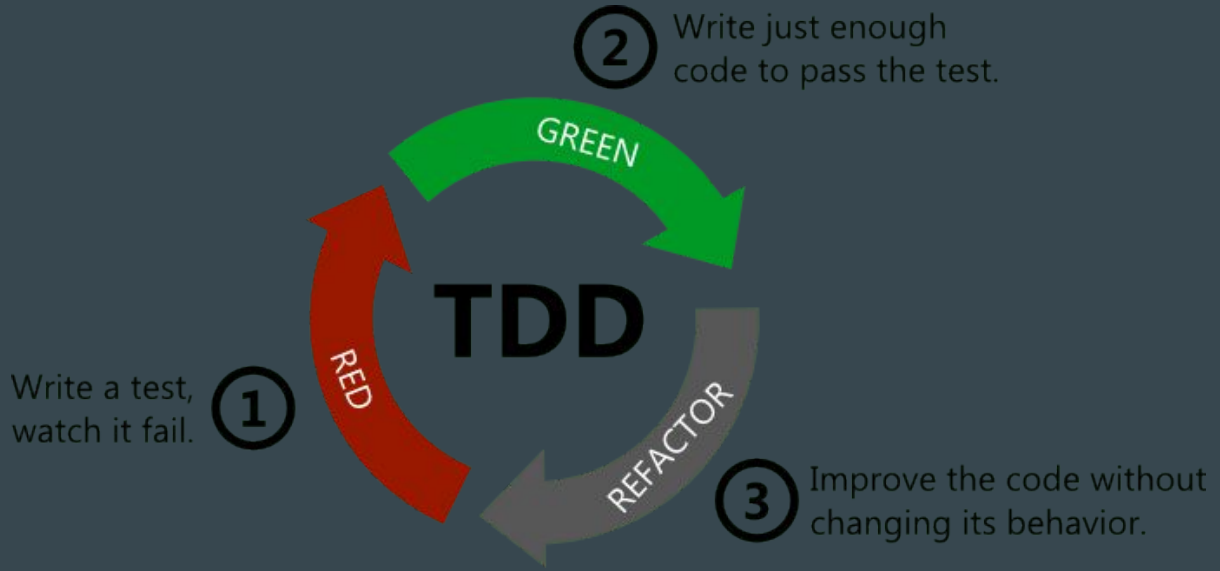
Flumine: A Change Impact Analysis Tool For Typescript

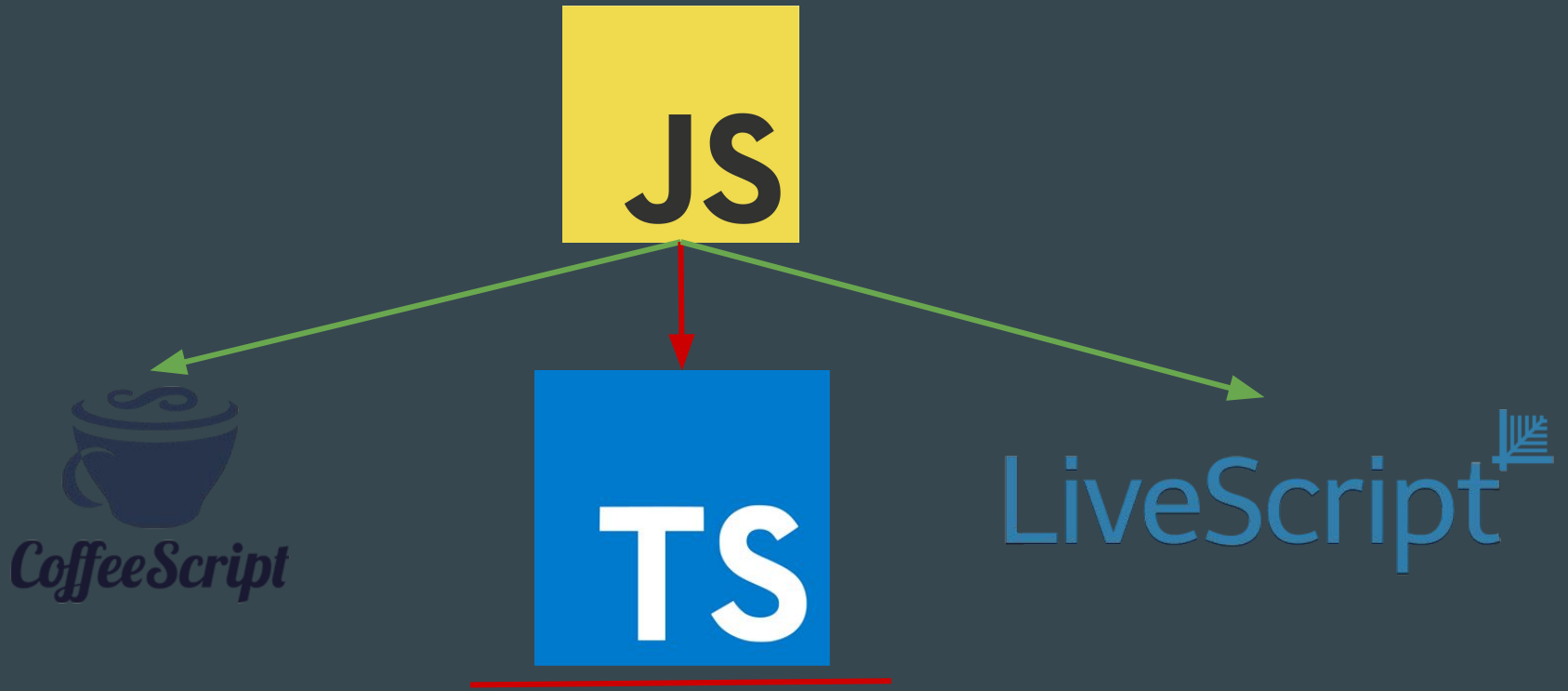
Nicolás Larrañaga Cifuentes - Andrés Mauricio Rondón Patiño

Introducción

Proyecto Grande + Etapas avanzadas de desarrollo + refactoring = Desastre

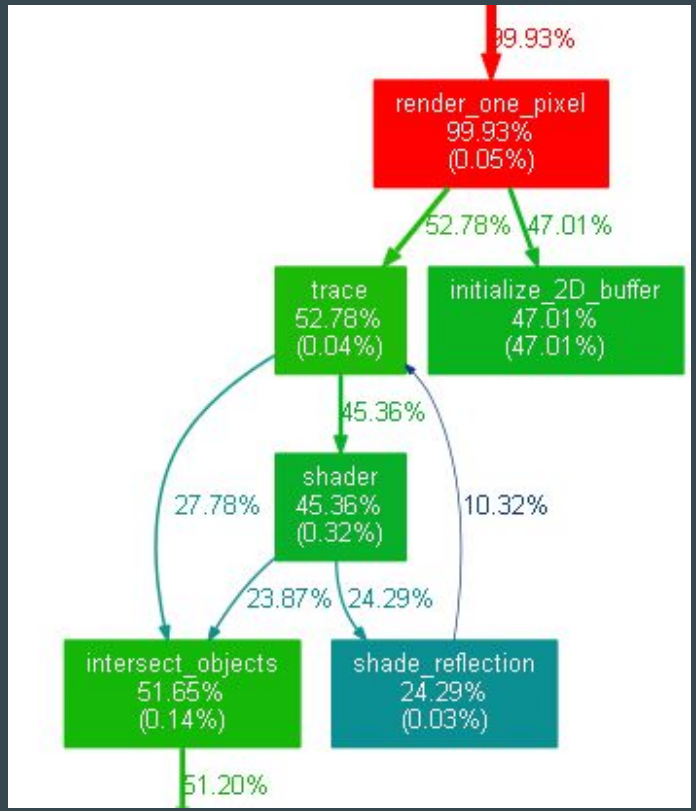


Typescript

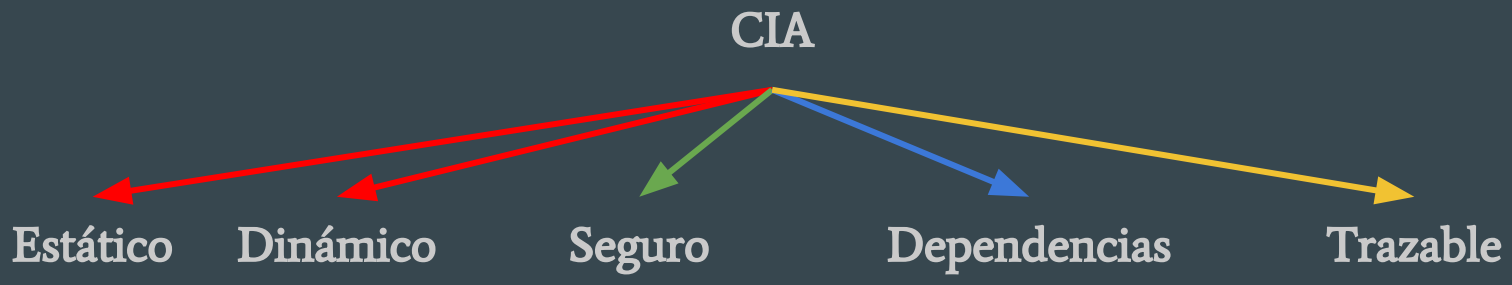


Change Impact Analysis

- ¿Cuales funciones se deben ajustar?
- ¿Cuales variables?
- ¿Cuales líneas de código?

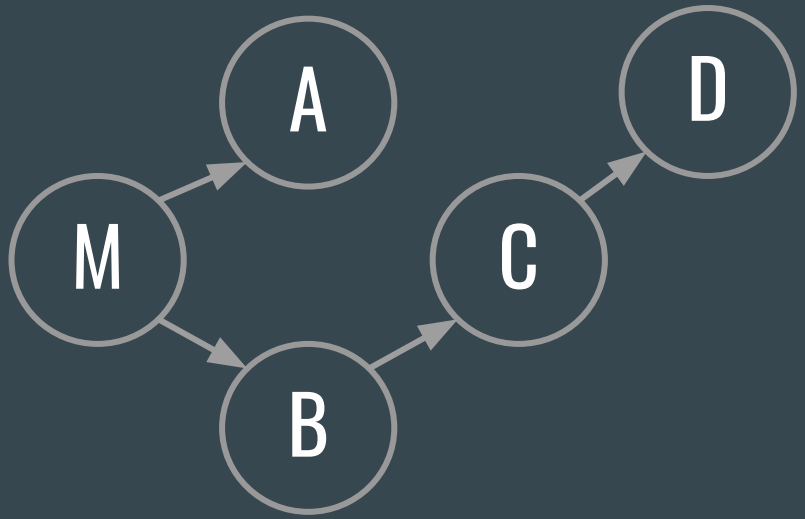


Change Impact Analysis



Algoritmos de Análisis Dinámico

PathImpact EAT CoverageImpact



Me	Ae	Ar	Ae	Ar	Be	Ce	Cr	Br	Be

PathImpact

1) Recorrido hacia adelante añadiendo cada método que tenga un Return sin padre
2) Recorrido hacia atrás haciendo emparejamiento de returns con métodos, lo que sobre es el impact set.

CoverageImpact

1) Crear Bitset con métodos alcanzados
2) incluir cada método que aparezca en la prueba.

M	A	B	C	D
1	1	1	1	0

PathImpact

Seguro

Lento

Costoso en memoria

CoverageImpact

Rápido

Ligero en memoria

No seguro

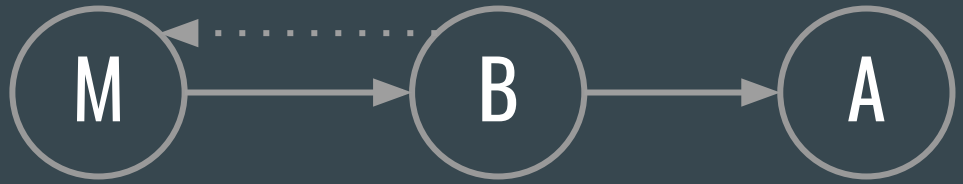
Información Contaminada

Flumine

- Es necesario un algoritmo eficiente, pero que ofrezca un alto nivel de seguridad, que funcione sobre métodos.

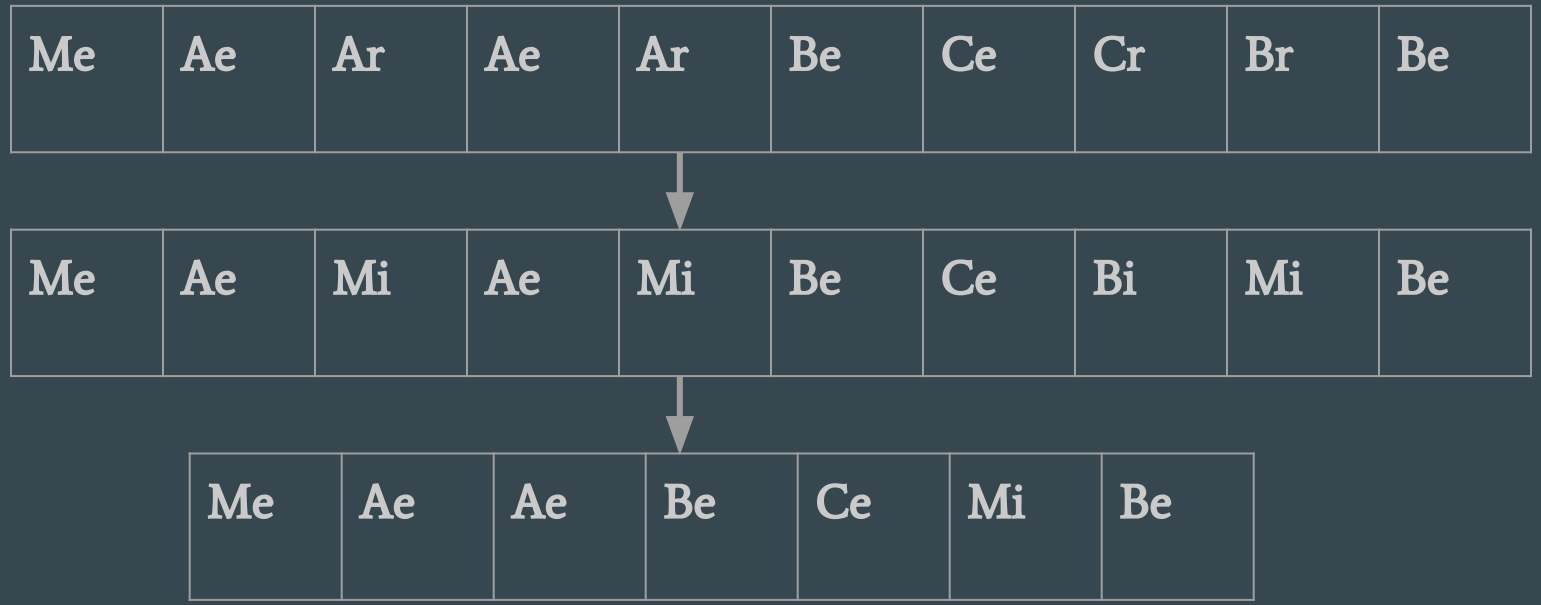
Execution After Sequences

Relación Execute After:

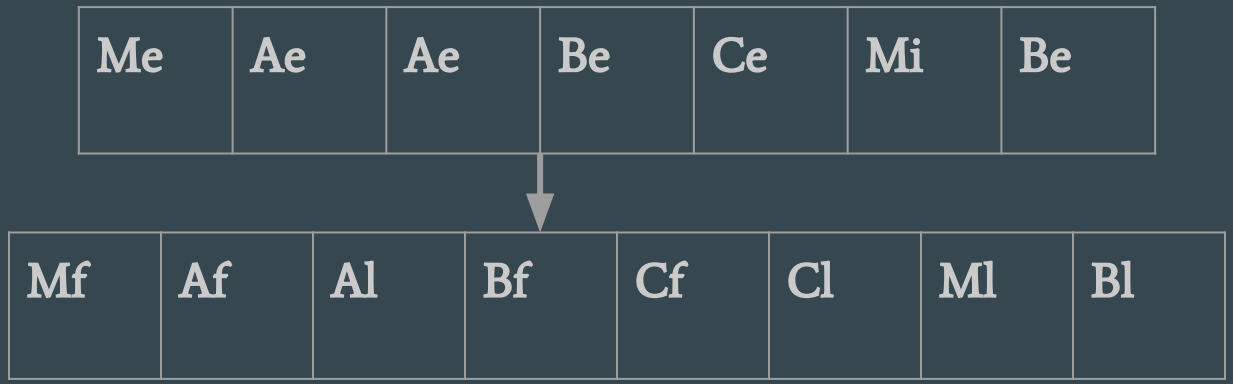


Execution After Sequences

Returned into X: Xi



Execution After Sequences



Algoritmo

L últimos eventos **F** primeros eventos **C** contador

1) Inicializar L y F como vacías, C en 0
2) Para cada entrada a un método si F[M] es vacío, iniciarlo con el tiempo actual.
3) Actualizar L[M] con el tiempo actual
4) Si se retorna de una función actualizar L[M] con el tiempo actual.

Para encontrar el Impact Set, basta con mirar los métodos que tengan F[M] < L[Changed]

Implementación en Typescript

- API Compilador de Typescript.
- Hooks
- **createImportDeclaration**
- Hacer uso de tests suites de cada proyecto para el análisis dinámico.

Ejemplo

https://github.com/larranaga/flumine

Conclusiones y Trabajo Futuro

- El desarrollo de nuevas herramientas como Flumine puede ser potenciado por las buenas prácticas de la industria.
- El desarrollo de nuevos lenguajes orientados a objetos necesita del desarrollo de herramientas que soporten su uso.
- El uso de algoritmos dinámicos en CIA es robusto y provee de un buen apoyo a los desarrolladores.

- Flumine sigue siendo un trabajo en proceso, existen casos que deben ser examinados a mayor detalle.
  - Arrow Functions
  - Manejo de excepciones
  - Higher order functions
- Una opción de análisis estático puede dar acceso a proyectos que no cuenten con extensos casos de prueba.

Referencias

[1] M. Usman, E. Mendes, F. Weidt, and R. Britto, “Effort estimation inagile software development,” inProceedings of the 10th InternationalConference on Predictive Models in Software Engineering - PROMISE’14, (New York, New York, USA), pp. 82–91, ACM Press, 2014.

[2] Robert C. Martin, “Clean code,” 2009.[3] B. Tanveer, L. Guzm´an, and U. M. Engel, “Effort estimation inagile software development: Case study and improvement framework,”Journal of Software: Evolution and Process, vol. 29, p. e1862, nov2017.[4] B. Tanveer, A. M. Vollmer, and U. M. Engel, “Utilizing ChangeImpact Analysis for Effort Estimation in Agile Development,” in201743rd Euromicro Conference on Software Engineering and AdvancedApplications (SEAA), pp. 430–434, IEEE, aug 2017.

[3] B. Tanveer, L. Guzm´an, and U. M. Engel, “Effort estimation inagile software development: Case study and improvement framework,”Journal of Software: Evolution and Process, vol. 29, p. e1862, nov2017.

Referencias

[4] B. Tanveer, A. M. Vollmer, and U. M. Engel, “Utilizing ChangeImpact Analysis for Effort Estimation in Agile Development,” in201743rd Euromicro Conference on Software Engineering and AdvancedApplications (SEAA), pp. 430–434, IEEE, aug 2017.

[5] J. Buckner, J. Buchta, M. Petrenko, and V. Rajlich, “JRipples: ATool for Program Comprehension during Incremental Change,” in13th International Workshop on Program Comprehension (IWPC’05),pp. 149–152, IEEE.

[6] B. Meyer, “Object-Oriented Software Construction SECOND EDI-TION,”[7] B. Li, X. Sun, H. Leung, and S. Zhang, “A survey of code-basedchange impact analysis techniques,”Software Testing, Verification andReliability, vol. 23, pp. 613–646, dec 2013.

[7] B. Li, X. Sun, H. Leung, and S. Zhang, “A survey of code-basedchange impact analysis techniques,”Software Testing, Verification andReliability, vol. 23, pp. 613–646, dec 2013.

Referencias

[8] J. Law, G. R. S. R. Engineering, undefined 2003, and undefined 2003,“Incremental dynamic impact analysis for evolving software systems,”ieeexplore.ieee.org.

[9] T. Apiwattanapong, A. Orso, and M. J. Harrold, “Efficient and precisedynamic impact analysis using execute-after sequences,” inProceed-ings of the 27th international conference on Software engineering,pp. 432–441, ACM, 2005.

[10] L. Badri, M. Badri, and D. St-Yves, “Supporting predictive changeimpact analysis: a control call graph based technique,” in12th Asia-Pacific Software Engineering Conference (APSEC’05), p. 9 pp., IEEE,2005.

[11] S. A. Bohner and R. S. Arnold,Software change impact analysis.IEEE Computer Society Press, 1996.

GRACIAS