



CHEMBIOINFO-03: Chem-Bioinformatics Congress München, Germany-  
Cambridge, UK-Chapell Hill, Durham, USA, 2021

## Software Testing Summary for Chemoinformatics

Shan He <sup>a</sup>, Paula Carracedo <sup>b</sup>

<sup>a</sup> Department of Organic and Inorganic Chemistry, Faculty of Science and Technology  
University of Basque Country UPV-EHU, 48940, Leioa, Basque Country, Spain

<sup>b</sup> Department of Computer Science and Information Technologies, Faculty of Informatics,  
University of Coruña UDC, 15071, A Coruña, Spain.

### Resumen

Las pruebas de software son investigaciones empíricas y técnicas que tienen como misión proporcionar a los usuarios información sobre la calidad del producto o servicio de software sometidos a prueba. <sup>1</sup>

Las técnicas de prueba incluyen el proceso de ejecutar un programa o aplicación con la intención de encontrar fallos y verificar que el producto de software sea apto para su uso. Además, estas pruebas implican la ejecución de un componente de software o un componente del sistema para evaluar una o más propiedades de interés. En general, estas propiedades informan hasta qué punto el componente o el sistema sometido a prueba:

- Cumple con los requisitos que guiaron su diseño y desarrollo
- Responde correctamente a todo tipo de entradas
- Realiza sus funciones en un plazo aceptable
- Se puede instalar y ejecutar en los entornos previstos
- Logra resultados generalmente que desean sus grupos de interés<sup>2</sup>

Existen diversas clasificaciones de pruebas de software que varían en función del modo de ejecución con o sin aplicación, el contenido que se verifican, nivel de prueba etc.

Según el modo de ejecución se encuentran las pruebas estáticas (para verificar) y dinámicas (para validar). La primera es el tipo de prueba que se

### Abstract

*Software tests are empirical and technical investigations whose mission is to provide users with information about the quality of the software product or service under test.*

*Testing techniques include the process of running a program or application with the intention of finding bugs and verifying that the software product is suitable for use. Additionally, these tests involve running a software component or a system component to evaluate one or more properties of interest. In general, these properties report the extent to which the component or system under test:*

- *Meets the requirements that guided its design and development*
- *Responds correctly to all types of entries*
- *Performs their duties within an acceptable time frame*
- *Can be installed and run in intended environments*
- *Achieves results generally desired by its stakeholders*

<sup>1</sup> Kaner, J.D. Exploratory Testing. Doctoral thesis, Stanford University, November 2006.

<sup>2</sup> Chairperson, J. R. Anonymous IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610. 12-1990, 1990*, 1-84.

realizan sin ejecutar el código de la aplicación. Por ejemplo, son las revisiones, recorridos o inspecciones. Mientras que el segundo, conlleva la ejecución de código programado con un conjunto determinado de casos de prueba. Las pruebas estáticas a menudo son implícitas, como la corrección de pruebas, además cuando las herramientas de programación/editores de texto verifican la estructura del código fuente o los compiladores (precompiladores) verifican la sintaxis y el flujo de datos como análisis de programa estático. Las pruebas dinámicas tienen lugar cuando se ejecuta el programa. Pero dichas pruebas pueden comenzar antes de que el programa esté completo al 100% para probar secciones particulares de código y se apliquen a funciones o módulos discretos.<sup>3,4</sup>

En función de los niveles de prueba, en términos generales, hay al menos tres niveles de prueba: prueba unitaria, prueba de integración y prueba del sistema.<sup>5</sup> Sin embargo, los desarrolladores pueden incluir un cuarto nivel, llamado pruebas de aceptación que sirven para asegurar que el software cumple las expectativas.<sup>6, 7, 8</sup>

#### **Pruebas unitarias**

Las pruebas unitarias tienen como objetivo garantizar el buen funcionamiento de una sección específica, así como aumentar la calidad y la eficiencia del proceso de desarrollo general de un software.<sup>9</sup>

#### **Pruebas de integración**

Es un tipo de prueba de software cuya misión es verificar las interfaces entre los componentes con un diseño de software. Los componentes de software pueden integrarse de forma iterativa o todos en conjunto. Normalmente, la primera es considerada una mejor práctica ya que permite localizar y solucionar los problemas de la interfaz de forma más rápida.<sup>10</sup>

*There are various classifications of software tests that vary depending on the execution mode with or without application, the content that is verified, test level etc.*

*Depending on the execution mode, there are static tests (to verify) and dynamic (to validate). The first is the type of tests that are performed without running the application code. For example, they are reviews, tours or inspections. While the second involves the execution of code programmed with a specific set of test cases. Static tests are often implicit, such as proofreading, in addition when programming tools / text editors check source code structure or compilers (precompilers) check syntax and data flow as static program analysis. Dynamic tests take place when the program runs. But such tests can begin before the program is 100% complete to test particular sections of code and apply to discrete modules or functions.*

*Depending on the test levels, generally speaking, there are at least three test levels: unit test, integration test, and system test. However, developers can include a fourth level, called acceptance testing, to ensure that the software meets expectations.*

#### **Unit tests**

*Unit tests are intended to ensure the proper functioning of a specific section, as well as to increase the quality and efficiency of the overall software development process.*

<sup>3</sup> Canning, R.G; Couger, J.D. *The Art Of Software Testing*; John Wiley & Sons, Inc, Toronto, 1979.

<sup>4</sup> Gragam, D; Van Veenendaal, E.; Evans, I. *Foundations of Software Testing: ISATQB Certification*; Course Technology Cengage Learning, 2008.

<sup>5</sup> Bourque, P.; Fairley, R.D. *Guide to the Software Engineering Body of Knowledge*; IEEE Computer Society, 2014.

<sup>6</sup> Lewis, W. E. *Software testing and continuous quality improvement*; Auerbach publications, 2004.

<sup>7</sup> Lewis, W.E. Waterfall Testing Review. *Software Testing and Continuous Quality Improvement*. Dobbs, D., Veerapillai, G., Eds., CRC Press, New York, 2016; Vol. 3, pp 92-96.

<sup>8</sup> Machado, P.; Vincenzi, A.; Maldonado, J.C. Chapter 1: Software Testing: An Overview. *Testing Techniques in Software Engineering*. In Borba, P., Cavalcanti, A., Sampaio, A., Woodcook, J., Eds., Springer Science & Business Media, Recife, 2007; pp 13-14.

<sup>9</sup> Binder, R. V. *Testing Object-Oriented Systems: Objects, Patterns, and Tools*; Addison-Wesley Professional, Toronto, 1999.

<sup>10</sup> Beizer, B. *Software Testing Techniques*; Van Nostrand Reinhold, New York, 1990.

### **Pruebas de sistema**

Las pruebas de sistema se emplean para confirmar que el sistema cumple con sus requisitos. Por ejemplo, una prueba de este tipo puede implicar probar una interfaz de inicio de sesión, luego crear y editar una entrada, además de enviar o imprimir resultados, seguido de un resumen de procesar o eliminar las entradas y luego cerrar la sesión.<sup>11</sup>

### **Pruebas de aceptación**

Las pruebas de aceptación incluyen cuatro niveles de prueba, los cuales son pruebas de aceptación del usuario, pruebas de aceptación operativa, pruebas de aceptación contractual y reglamentaria y pruebas alfa y beta. La aceptación operativa se utiliza para llevar a cabo la preparación operativa (prelanzamiento) de un producto, servicio o como parte de un sistema de gestión de calidad. Por otro lado, las pruebas de aceptación contractual se realizan en base a los criterios de aceptación del contrato. Mientras que las pruebas de aceptación regulatorias se proceden en función de las regulaciones relevantes para el producto de software. Por otra parte, las pruebas alfa son pruebas operativas simuladas o reales realizadas por usuarios/clientes potenciales o un equipo de prueba independiente de los desarrolladores.<sup>12</sup> Finalmente, las pruebas beta son pruebas que vienen después de las pruebas alfa. Estas pruebas se lanzan a una audiencia limitada fuera del equipo de programación conocidos como probadores beta. Con lo cual, el software se distribuye a grupos de personas para someter a las pruebas adicionales con el fin de asegurar que el producto tenga pocos fallos o errores.<sup>13</sup>

### **Integration testing**

*It is a type of software test whose mission is to verify the interfaces between components with a software design. The software components can be integrated iteratively or all together. The former is typically considered a best practice as it allows for faster troubleshooting of the interface.*

### **System tests**

*System tests are used to confirm that the system meets your requirements. For example, such a test might involve testing a login interface, then creating and editing an entry, as well as submitting or printing results, followed by a summary of processing or deleting the entries, and then logging out.*

### **Acceptance Tests**

*Acceptance testing includes four levels of testing, which are user acceptance testing, operational acceptance testing, regulatory and contractual acceptance testing, and alpha and beta testing. Operational acceptance is used to carry out the operational preparation (pre-launch) of a product, service or as part of a quality management system. On the other hand, the contractual acceptance tests are carried out based on the acceptance criteria of the contract. While regulatory acceptance tests are proceeded based on the relevant regulations for the software product. On the other hand, alpha tests are simulated or real operational tests performed by potential users / customers or a test team independent of the developers. Finally, beta tests are tests that come after alpha tests. These tests are released to a limited audience outside of the programming team known as beta testers.*

<sup>11</sup> Xuan, J.F.; Monperrus, M. Test case Purification for Improving Fault Localization. *Proceedings of the 22<sup>nd</sup> ACM SIGSOFT International Symposium on Foundations of Software Engineering-FSE*. 2014, 52-63.

<sup>12</sup> Standard Glossary of Terms used in Software Testing [online]; International Software Testing Qualifications Board, Posted July 2010. <https://astqb.org/assets/documents/Glossary-of-Software-Testing-Terms-v3.pdf> (acceso 4 Sep, 2021)

<sup>13</sup> End of the Software Release Cycle [Online]; O'Reilly Media, Posted Sep 30, 2005. <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=4> (acceso 4 Sep, 2021)

	<p><i>Whereupon, the software is distributed to groups of people for further testing to ensure that the product has few bugs or errors.</i></p>
--	---