**The UML Design Modeling paper**

Be Huynh

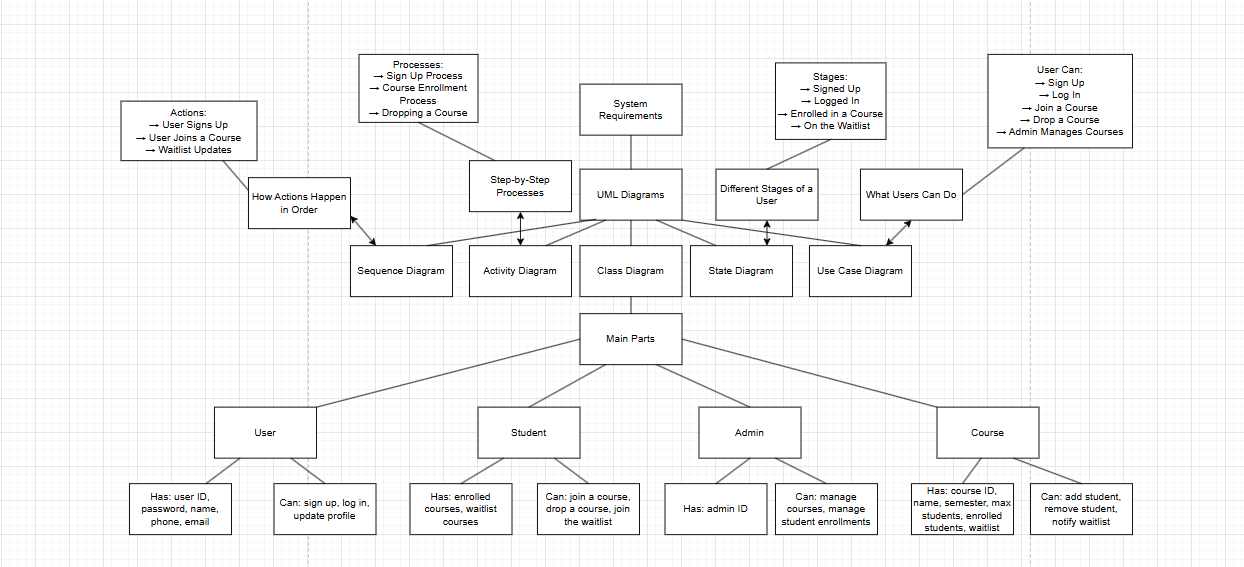
University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Professor Joseph Rangitsch

February 10, 2025

In software development, testing is a very important step to ensure that the software works properly and meets the needs of the users. The main goal of testing is to find any issues or bugs early so that they can be fixed before the software is released. There are different types of testing, and each one checks a different part of the software. The four main types of testing are component testing, integration testing, system testing, and acceptance testing. Each level of testing has its own job to make sure the software works smoothly from start to finish.Component testing, also known as unit testing, is the first level of testing. It focuses on the smallest parts of the software, like a single function, button, or piece of code. The goal is to make sure that each part works correctly on its own. For example, if there is a calculator feature in the software, the developer might test if the “add” function correctly adds numbers together. This helps find and fix issues with individual parts before they are combined with other parts. If something is wrong at this level, it is easier to fix because it is isolated to just one small piece.Once the individual components have been tested, the next step is integration testing. This type of testing checks if different parts of the software work together properly. For example, in an online shopping website, integration testing would check if the shopping cart is correctly connected to the payment system. If a user adds an item to their cart and goes to the checkout, integration testing makes sure that the cart and the payment system work smoothly together. This testing ensures that the parts, when combined, communicate correctly and that there are no issues when they interact with each other.System testing looks at the entire software as a whole. This is when the software is tested to see if everything works as expected in real-life situations. For example, system testing would check if a user can log in, browse the website, add items to the cart, and make a purchase without any problems. It makes sure all the parts of the software are working together and that the system performs well under normal and different conditions. System testing ensures that the software does what it’s supposed to do when used by real users.The final level of testing is acceptance testing. This is the step where the software is tested by real users or stakeholders to see if it meets their needs and requirements. Acceptance testing is done to make sure the software is not only functional but also easy to use and provides the features that users expect. For example, a group of users might test the software to see if it’s easy to navigate, if all the features work as expected, and if they are satisfied with the overall experience. If anything is wrong or doesn’t meet the user’s expectations, it can be fixed before the software is released to the public.In conclusion, testing is a key part of software development because it ensures that the software works correctly and meets the needs of the users. By testing the small parts first (component testing), then checking how the parts work together (integration testing), testing the whole system (system testing), and finally making sure the software meets user needs (acceptance testing), developers can find and fix problems before the software is released. This helps make sure that the final product is reliable, user-friendly, and ready for use. Testing at each of these levels is essential for creating high-quality software that performs well and is easy for users to interact with. In the end, a well tested software product not only reduces the risk of failures but also builds trust with users, ensuring they have a positive experience with the software.



[UML diagrams](https://viewer.diagrams.net/?tags=%7B%7D&lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&title=UML%20diagrams#R%3Cmxfile%3E%3Cdiagram%20id%3D%22i_AC_6rn9npSOfA9vU98%22%20name%3D%22Page-2%22%3E7Z1bc5s4FIB%2FjR%2BzA8LXx8TONu0228x6Mm33ZUc1ik3LrUKO7f31KxnETdjGRljKlplkAgIkcc6no3OOxKRnTb3tOwzD1WNgI7cHDHvbs2Y9AMDYNOgfVrKLSybmJC5YYseOi8ysYO78i5LC5Lnl2rFRVLiRBIFLnLBYuAh8Hy1IoQxiHGyKt70EbrHVEC6RUDBfQFcs%2FezYZBWXjgdGVv6AnOWKt2wayRUP8puTgmgF7WCTK7Lue9YUBwGJj7ztFLlMeFwu8XO%2FH7iadgwjn9R54PbH5suH8NM%2F87vn%2FuDPd39%2FBX883iS1vEJ3nbzwI3R8WvIEMYmSjpMdlwYO1r6NWIVGz7rbrByC5iFcsKsbqn9atiKeS89MephUjTBB24N9NlNJUIRQ4CGCd%2FSW5AGLCy%2Bhxxon55tMFya%2FZ5XTwzApg4n6l2nVmYToQSKkMwQGBIHdLojz6hDW65kDlxh6ysVmTkpiM1SLzRLE9hyhHuvC0KVt30UhZNQtXBjFTQ9%2FrtnAuFu536ObH2i3CbCdldKjJfs7hayWuArar7iW%2BJJG6hhop46%2BoI45gQRpJLP%2BQDeZDQSZNYR3jn6ukb94CwDrpouhbF1M9%2FdrrwhhPlSuiZFo2B8%2FZhLTz4VIPUtlIhtLNyS7iCDvML1%2FUUPjYORRiemnD1O5PiaCPqZUfNYtk5OzZApZhz0wbe6tuMHysJIa1s2c9gNVS%2Bj5OrT37sGhzoc4eHEoUarhuukX4Roojxc43Tm6HlgjtxKUEiF8WCXvZ1I0H9LbkytN6SK7EIkN%2BNA7DBarPFwFPooPkQcdVz1k6RSiD2ViHD8N1jhSPyJHpRHZnyiXlRjCNxwjt7Z3xPyqDqSG2ilATAbMydpm76NaWMDQTlhiqP68t%2FuqjaBpaicqMUKfExTeNBzd33YHhvYNq51W%2BISDBYoipN63Luf9gPLw0BQjdZYvDfzYBYrlajuvqbm8B73xsDcBzT2X56Pu0Zx69lRhxnPI7%2FmGC9541c%2B1OvchcPadg%2FSXz%2BOXdPIzdIjrRGT%2FnsyDz6c48lJXDe5NukrCo0KrgtxhBbmj1sgVMxsPwYYWcH6B8QDDEPmtxm5Nq%2F6E7SOgqdb6qGSugHJzJeZm0lxAQ01A%2B3AMRZWQ%2BD8sysHIC15RubRh835AnJdDMxkTeWoqFEMxLiWbNQiuxARRGsIvEuuchttxOMuOIuShiLDRx848uM00GiXhrI8D16WyLV%2FQRhkj7SJd3liVMmAch%2B11oTrqAtoJriJFwE2bB322DYLjHHFkk9LUDiXE6pFIHmhnKICYWZA1e3wPjjkJMGeJJMwVNg7CqzV2%2FM3ICulkD8FIO%2BbEZEpqD3PzS2FgZ%2BLMylVHrvrZSzHxkgX6J2PXebxw9JzLD5wRwgmBHA8CjfvUAMupeEaHeuj4y15VsKlZkFhKZI4GIiH9CkLGrREiRoitZgX4XFIXvHOokJfZ%2BNjiIuf7wzOFmpf9cHJaPjt9I7V%2Fs5Mzudr%2BHV85MR73HmiUdlPX9NW4tGxQZZq457Aq7hZuyzaJiYzPK7hPBFIbEyXWpDlfga65pWEpOlC%2F7QeIaYQ52eNdy6jv3bj6Zl2qQV%2BiI7kr1TY996b3mcPbYkr2fLMpTxmfTsZKWc5fS1NZjtrHFabSqhiWk7aGJW8sNyxnzssLwrGPnQzRxooLXo4D1abnqDoXNi6qXL0ptsRcmCAl5Nu37PsheuazDUgFqRRFSOWAd1%2FYyW8Dfvo1f222LZzt%2BNnWIfFjo0Fy%2BjVpgB1nT7ET%2FlDcUWQLny2V1EFfhtqoBToiBr4ThkC8ROTY1HVAwTkFDir0x8swciFxXosdrlJq0sIT9apJjp%2FypgyrxEX8oslTGRpiReV97OXQMBaEUNGesfS1G2An5gfbwM6sjV2eOuNa1HHtnaZu0lEngzoxkaXW2Glu64yOOhnUVX0uJYE68yLm8nYu5a995moiF%2B%2Fx65BrilzVV2EyDN0lyJmaI9d5dFKQq1oC%2BAUDibrUgY46GdRVfUEohTrwlgIJ%2Fh3Baew66mRQJ%2Bav27B1ukNXk7lhx5wE5vh6%2Fq%2FN3LAmc6OOORnMXSc5rDlzo5rMjTvmZDAH2mIOvKlAoi51XcJECnXi3tE2LN0lKRPtiLM64mQQd521iPpzK1ATRAxqUgc6QycFu9bSdJclTBQlh%2FmmwNPYdZGEDOx48%2FKxMwfjwgz722RsnWKPnT0h7NC3Qvi4v3c1Q1jXDnY8SuGxtcj20ryxGupAR901qWstyhhdRl1h8gXXW5qtG2qYXe5YCnfn7UDZ77h1FlTnEe0YEYv%2Fbxm%2BulYwDlY6HJvieN7uFAk4FrdK5aylplFxXWfQ7OZlKUCeFxSrtI9q3MR%2BTR6tbseoFB7P29Xy6%2FFYd7sL6NbkZPA4BGfxeHHcUn9NDqjZ3FfXDpqdHZTCnV6rcsr2lNbeVNqZOynY6faZkCLs6lq77uM0KdS19JnQG9vKDGpvZe72IUjBrq1Phd6YsasdUoAOOxnYtbYPoW9ZZ5m744vBajbFTOrawO57juMw0tPs38rGt2f%2FnNe6%2Fw8%3D%3C%2Fdiagram%3E%3C%2Fmxfile%3E)

**References**

*Differences between the different levels & types of testing*. (2023, August 24). Reqtest. https://reqtest.com/en/knowledgebase/different-levels-of-testing/

GeeksforGeeks. (2024, February 25). *Levels of software testing*. GeeksforGeeks. https://www.geeksforgeeks.org/levels-of-software-testing/

Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning.