|  |  |  |
| --- | --- | --- |
| pampa |  |  |
| Filmina 1 | Presentacion | Presentation de integrates |
| Filmina 2 | Que es Agile manifesto | Agile testing is not just the same traditional testing but in sprints. The Agile approach must completely change the way the team thinks. About 2 years ago we created our version of a testing manifesto, as a quick summary of the mindset you should adopt when thinking about agile testing. We thought it was pretty cool 🙂 Apparently so did many others, and the slide we had has been retweeted and added to many presentations since then. |
| Filmina 3 | Testing durante | *Testing durante****SOBRE****testing al final: Priorizar la realización de tareas de testing desde el inicio hasta que se da por hecha****por sobre****realizar testing como una etapa al finalizar el sprint.* |
| Filmina 4 | Prevenir bugs | *Prevenir bugs****SOBRE****encontrar bugs: Incentivar la discusión y revisión grupal de las tareas, tener en claro que hay que hacer, como y porque****,****evitar todo tipo de suposiciones y trabajar realmente en equipo con el desarrollador para que los bugs se puedan evitar de antemano****por sobre****esperar a que el desarrollo este completo para revisar y estudiar el mismo, así como evitar discutir con el desarrollador sobre que se va a testear* |
| Gasty69 |  |  |
| Filmina 5 | Testing understanding | *discutir con el desarrollador sobre que se va a testear.*  *Entender lo que se esta testeando****SOBRE****verificar funcionalidad: La necesidad de entender que es lo que quiere el usuario, para que lo quiere y como él lo va a usar, para realizar pruebas que den valor agregado****por sobre****solo ver que las especificaciones se cumplen (ver que 1+1 sea igual a 2) en una tarea mas de checklist que de testing* |
| Filmina 6 | Construer el major sistema | *Construir el mejor sistema****SOBRE****romper el sistema: Poner el trabajo de equipo en generar el ambiente necesario para poder mejorar lo que se esta haciendo****por sobre****solo intentar romper el sistema.* |
| Filmina 7 | Responsabilidad de equipo | *El equipo es responsable de la calidad****SOBRE****el tester es responsable de la calidad: En agile la clave de entregar un producto con calidad es que el equipo se comprometa con todas las actividades y trabajen conjuntamente para generar valor, por eso es importante que la responsabilidad sobre la calidad sea del equipo****por sobre****que la calidad recaiga sobre un solo rol como es el del tester.* |
| Filmina 8 | Cambiar requerimientos | It can sometimes happen that management changes requirements or drops stories during a sprint, even though this is not encouraged in an [agile/Scrum framework](https://www.sealights.io/software-development-metrics/11-scrum-metrics-and-their-value-to-scrum-teams/). This means that work already half-done needs to be discarded or modified, which changes the scope of testing unexpectedly. |
| facu |  |  |
| Filmina 9 | Solucion1 | Testers should be able to react and modify their processes according to changing conditions, because in agile projects, change is common. When requirements change, testers should share as much information as they can about what tests they have conducted, and which areas of the application have not been tested yet. This can help the team understand how to make the required changes to the sprint without hurting the quality of the release. |
| Filmina 10 | info insuficiente | Product owners, who are responsible for developing [user stories](https://www.mountaingoatsoftware.com/agile/user-stories), may have an idea about a new feature but may not be aware of the specifics. This means they can’t write a good set of acceptance criteria. If there is missing information about requirements, testers can’t build comprehensive test cases. |
| Filmina 11 | Solucion 2 | Testers do not need in-depth requirements to begin testing, they can begin by coming up with high level scenarios that test the idea of the story, confirming them with the product owner. Testing can be done without the complete details about a feature. You can create high level test scenarios, even when particulars change. |
| Filmina 12 | Testing continuo | Testing is not restricted to one part of the development process, rather it’s an[ongoing activity](https://www.sealights.io/webinars/continuous-testing-the-future-of-qa-funnel-form/) that starts before the development phase. This creates a major challenge because testers are expected to start building tests for features before coding has even started, or while coding is taking place. |
| Gaston V |  |  |
| Filmina 13 | Solucion 3 | To make life easier for testers, user stories in the backlog should be expanded during sprint planning. Testers, developers and product owners should jointly define the details of each story and then write effective acceptance criteria.  The team should ensure that each story has sufficient acceptance criteria and that the context of the story is universally understood before work on development begins. This makes it possible to start creating tests early on, which can be implemented when the code for the feature is complete. |
| Filmina 14 | Falta de comunicacion | If communication between developers, testers and the product owners is lacking, agile testing will simply not work. |
| Filmina 15 | Solucion 4 | Direct communication within the team should be strongly encouraged. Developers, testers and product owners should talk face-to-face on a regular basis to ensure everyone is on the same page. Scrum ceremonies such as stand-up meetings, [sprint planning](https://www.sealights.io/software-development-metrics/sprint-planning-101-everything-you-need-to-know-to-plan-the-perfect-sprint/), and retrospectives are instrumental in creating a common understanding of the sprint scope and goals. |
| Filmina 16 | Regretion cycles | Developers frequently and continuously add features to the product. This can cause regressions in previous features. Testers use [regression tests](https://www.sealights.io/test-metrics/regression-testing-in-agile-concepts-challenges-and-strategies/) to identify this problem and overcome it, but manual regression testing is impractical in a fast-paced agile environment.  Another challenge is that modern web applications behave differently when viewed on different devices or browsers. This creates a complex matrix of compatibility testing scenarios, which need to be tested to ensure that the application functions correctly for all users. |
| tami |  |  |
| Filmina 17 | Solucion 5 | Agile testers rely on [automation](https://www.sealights.io/wp/untangling-automated-testing-and-continuous-testing/). They use unit testing to ensure recent changes have not broken the code, and tools like Selenium and JMeter to verify there is no regression in basic functionality. Testers can use Docker or Selenium Grid to manage and run their automated test, in parallel on various browsers and machines. |
| Filmina 18 | Skills tecnicas | Testers who work in an agile environment need to be technically savvy, helping developers with API testing, integration testing, and scripting UI automation checks with Selenium or similar frameworks. Testers with an exploratory or manual testing background entering the world of agile will encounter a steep learning curve. |
| Filmina 19 | Solucion 6 | Testers can and should learn programming or scripting languages such as [Javascript](https://www.sealights.io/test-metrics/javascript-code-coverage/) and Ruby. Testers who are familiar with programming but lack practical experience can ask for help from developers. Testers can also learn automated testing tools like Selenium tool and JMeter.  For specialized testing areas, such as performance, security, or compliance testing, teams should have dedicated testers with the relevant professional background, or leverage consultants with deep experience in these areas. |
| Filmina 20 | Final + fuentes | Cerrar la presentacion agradeciendo por el espacio señalar las fuentes y esperar preguntas |