

① La crisis del software es un periodo de la historia del SW, en la cual ocurrieron muchísimos fracasos en llevar proyectos a productos terminados. Existió la necesidad de SW, pero aún no existen metodologías viables para desarrollar de manera eficiente, ni técnicas adecuadas para estimar. Principalmente ocurrió en los 80, pero se extendió hasta los 90. Se desperdició muchísimo dinero y tiempo. Su impacto positivo fue crear la necesidad de nuevas metodologías y frameworks de desarrollo, lo que llevó al manifiesto agil y a lo que conocemos como agilidad.

② Los equipos de desarrollo de más de 9 personas poseen muchas conexiones internas (36, para ser exactos). El trabajo en equipo inspira a alcanzar la excelencia, el auto perfeccionamiento y motiva a los miembros, pero el exceso de relaciones lo vuelve complejo, y ralentiza la toma de decisiones (ponerse de acuerdo cuesta más). ~~the~~

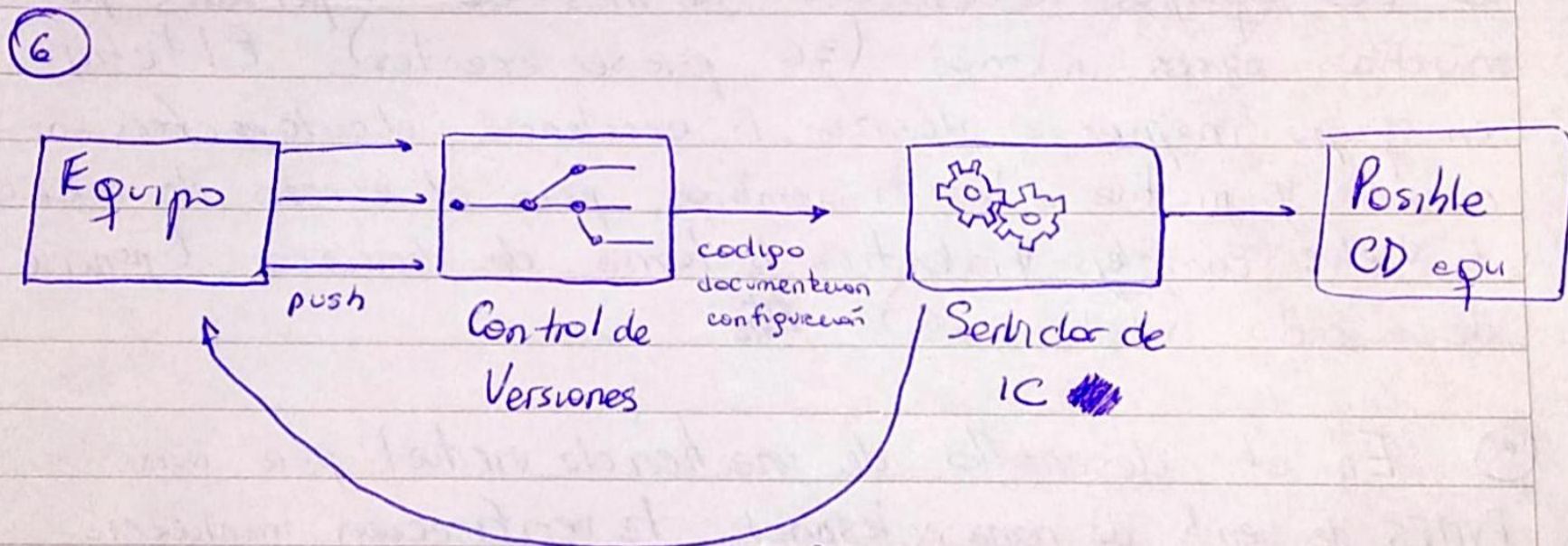
③ En el desarrollo de una tienda virtual para una PYMES de venta de ropa artesanal, la verificación implicaría comprobar que la página funcione bien, y la validación, que mi cliente pruebe el producto y me confirme que es equivalente a lo que tenía en mente cuando solicitó el trabajo.

④ En canary release, se despliega la nueva versión y se le dirige un porcentaje pequeño de la carga de trabajo. Se analiza el funcionamiento de la nueva versión mientras la anterior continúa trabajando. Cuando estamos convencidos que la nueva ~~función~~ versión funciona bien, dirigimos todas las solicitudes a la misma, bajando la anterior.



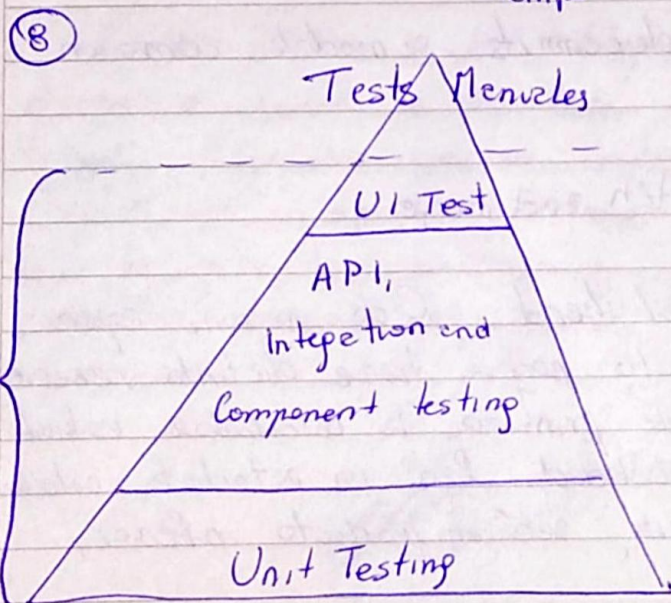
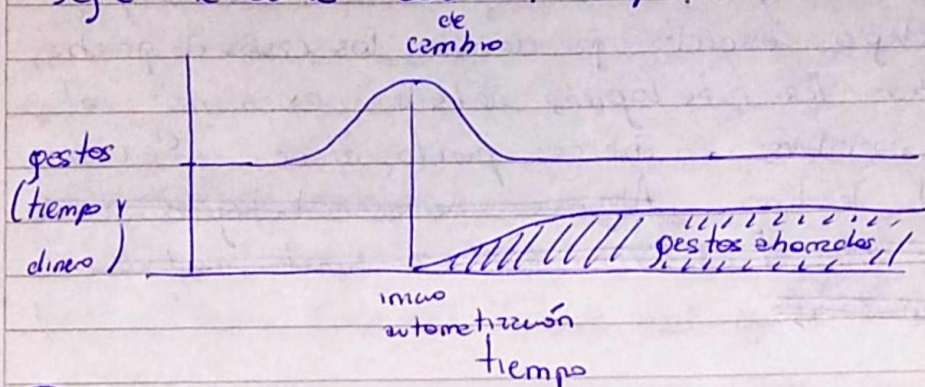
En cambio, en Blue Green, subimos la nueva versión y bajamos la anterior. Dirigimos desde ese momento todo el tráfico a la nueva versión. Podemos decir que la Canary release es más segura, dándonos una prueba de sistema extra, pero más difícil de implementar (hoy en día las herramientas hacen la mayor parte de este trabajo por uno, pero si requiere de más recursos).

④ Un caso de prueba es una situación (condiciones, parámetros, entradas, variables de sistema, etc) que es posible que ocurra en un sistema. Conociendo como debería responder el sistema frente a la misma, se puede diseñar una prueba para comprobar el ~~funcio~~ buen funcionamiento del sistema frente a dicha situación.





⑦ Al principio toma un poco de tiempo y dinero extra implementar test automáticos, pero es una inversión que reduce costos a largo plazo. Cuando estamos al final del proyecto, los test manuales (sin automatización implementada) son muchos y toman mucho tiempo. Además, al simplificar la ejecución del testeo, el mismo se realiza más a menudo, los errores se encuentran antes. Mantenemos baja la curva costos / tiempo.



Como recomiendo automatizar mucho a nivel de método / función / objeto, los llamados tests unitarios. Luego, desarrollar tests automáticos de API, de integración y de componentes en menor cantidad, lo cual prueba los servicios del sistema. Finalmente, hacer algunos tests de la interfaz. Estos tests



son mas dificiles de automatizar y ven a requerir cambios constantes. Al final, siempre vamos a tener tests manuales y exploratorios para las nuevas funcionalidades, donde se necesite el analisis de un ser humano frente a los resultados.

Cuando los resultados de estos tests manuales se vuelven mas predecibles y estables, algunos de los mismos pueden automatizarse y pasen a niveles inferiores de la piramide.

⑨ La prueba de caja blanca testing a nivel de código, usando el código escrito para derivar los casos de prueba, se busca comprobar los rutines lógicas del sw y examinar el estado de las variables en ciertos puntos clave. Se usan mucho en Unit testing. ~~Algunas de las estrategias donde mas se utilizan son el analisis de valores limite y el de particiones exponenciales.~~

⑪ En git, el historial de commits se modela como un grafo aciclic dirigido.

⑫ ② Pull combines fetch and merge

⑬ La calidad depende del cliente. Cada persona espera algo distinto de un producto porque tiene distintas necesidades. Ej: para un estudiante de primaria, la calculadora estandar de windows sera de alta calidad. Para un estadista calculando el progreso futuro del covid, sera un producto inferior, ineiciente y limitado.

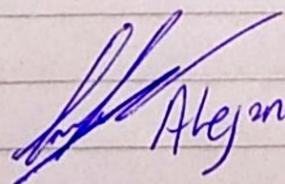
⑭ Una técnica de caja negra es la prueba de comparacion, donde comparamos los resultados de las nuevas versiones vs las anteriores, buscando asegurar la fiabilidad del sw. Solo se analizan los resultados



(14) La calidad del proceso es un conjunto de normas y métodos que un equipo sigue, y a su vez para edificar, suministrar, desarrollar, instalar y/o mantener, SW y servicios de soporte. La calidad del proceso afecta a la calidad del producto final.

(15) Deuda técnica buena: Alejandro debe terminar su historial de usuario por mañana, lo cual implique realizar un conjunto de operaciones con arreglos. Alejandro intentó aplicar vectorización pero las dimensiones de los arreglos no coinciden. Entonces, implementó la funcionalidad con un conjunto de bucles, sabiendo que es mucho más lento. En el siguiente sprint, Alejandro implementó vectorización y el método funciona 7 veces más rápido.

Deuda técnica mala: Alejandro debe terminar una US donde se conecte a una base de datos. Por cuestiones de tiempo, Alejandro crea un endpoint que no requiere autorización. Tras terminar el sprint, Alejandro siempre se dice a sí mismo que arreglará la autorización. Pero siempre se olvida. El producto termina desplegado con endpoints inseguros en producción.

 Alejandro Naezel