Indice

1. [Test Plan Identifier](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#1)
2. [References](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#2)
3. [Introduction](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#3)
4. [Test Items](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#4)
5. [Software Risk Issues](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#5)
6. [Features to be Tested](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#6)
7. [Features not to be Tested](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#7)
8. [Approach](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#8)
9. [Item Pass/Fail Criteria](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#9)
10. [Suspension Criteria and Resumption Requirements](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#10)
11. [Test Deliverables](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#11)
12. [Remaining Test Tasks](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#12)
13. [Environmental Needs](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#13)
14. [Staffing and Training Needs](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#14)
15. [Responsibilities](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#15)
16. [Schedule](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#16)
17. [Planning Risks and Contingencies](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#17)
18. [Approvals](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#18)
19. [Glossary](http://www.gerrardconsulting.com/tkb/guidelines/ieee829/main.html#19)

Plan de pruebas unitarias

# Identificador del plan de prueba

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Responsable(s) |
| 13/05/18 | 1.0 | Plan de pruebas del sistema CineTEC | Dennis Arias León  Óscar Cortés Cordero, Danny Xie Li |

# Referencias

[1]. O. Cortés, D. Arias, D. Xie. Documento de visión. 2018

[2]. O. Cortés, D. Arias, D. Xie. Prototipo del sistema, diseño a alto nivel. 2018

[3]. O. Cortés, D. Arias, D. Xie. Documento de especificación de requerimientos. 2018

# Introducción

# En el presente documento se especifica el plan de prueba que se va a aplicar en el sistema, para que el grupo de soporte y probadores puedan realizar validaciones, corrección de errores, verificación de datos el caso. Además, por medio de la realización del plan de pruebas se podrá tener los casos de éxitos o fracasos que se obtuvieron en el desarrollo del proyecto, por ejemplo, cuales casos de usos se pudieron completar exitosamente o los casos de usos que tuvieron dificultades durante el desarrollo del sistema. Al desarrollar el plan de pruebas se podrá detectar objetos faltantes o objetos innecesarios en los prototipos diseñados anteriormente, así se podrá asegurar tener una mejor calidad en el producto para satisfacer exitosamente los casos de usos que el cliente pidió.

Elementos de pruebas

En este apartado se mencionará los elementos que se someterá a prueba, las funcionalidades que se someterán a prueba serán entregadas al cliente satisfactoriamente.

# Funcionalidades:

# El sistema permitirá el registro y la actualización de películas por parte del administrador.

# Permitir bloquear y desbloquear usuarios para que el(los) administrador(es) puedan bloquear o desbloquear usuarios en el sistema.

# Se podrá realizar la consulta de películas por medio de actores, nombre de película, keywords, director de película y género, tanto en la parte del administrador como el usuario regular.

# Seleccionar películas como favoritas por parte del usuario regular.

# Permitir la visualización de recomendaciones de películas hechas por el sistema por parte del usuario regular.

# Permitir la visualización de películas seleccionadas como favoritas por medio del usuario regular.

* El sistema le permitirá al usuario regular comentar sobre cualquier película que se encuentra registrado dentro del sistema.
* Tanto el usuario administrador como el usuario regular el sistema le permitirá mostrar las películas que se encuentra registrado en el sistema.
* En el sistema permitirá la creación de usuarios como administrador o usuario regular.

# Problemas de riesgos en el software

Entrega de un producto a terceros: AL entregar el producto al cliente, el producto esta propenso a ser aceptado o no por los usuarios que van a usar el sistema por ejemplo el usuario regular o administrador lo que podría generar un gran impacto en el cliente ya sea positivo o negativo por parte de los usuarios que utilizan dicho sistema.

Nueva versión de interfaz del software: Al sacar una nueva versión de interfaz de software este podrá tener un gran impacto al cliente debido a que este podrá desconocer de cómo funciona el sistema, provocar confusiones a la hora de usar el sistema, puede provocar un impacto negativo.

Funcionalidades extremadamente complejas: El sistema puede ser demasiado complejo para lograr entender cómo funciona este, es un riesgo posible de desapruebo por parte del cliente.

Un problema de riesgo de software es presentar una documentación pobre de los cambios o módulos que se diseñaron anteriormente antes de entregar el producto, este dificultad la compresión por parte de grupos de trabajo que desee trabajar en mejoras, cambios o agregar nuevos casos de uso que se va a realizar en el sistema.

# Caracteristicas a ser probadas

En el siguiente listado se mostrará las caracteristicas que serán probadas, según una escala de riesgo: Alto: A, Medio: M, Bajo: B, donde Alto es u riesgo mayor de que no se presente la funcionalidad en el sistema y bajo es probable que se presente en el sistema.

* El usuario regular podrá buscar películas. B
* El usuario regular podrá seleccionar películas como favoritas. B
* El administrador podrá registrar y actualizar películas en el sistema. B
* El administrador tendrá la capacidad de bloquea y desbloquear usuarios. B
* Visualizar recomendaciones de películas por parte del sistema. A
* Usuario regular podrá visualizar películas marcadas como favoritas. A
* Usuario regular podrá comentar en las películas. M
* Tanto el usuario regular como el administrador podrá visualizar películas registrados en el sistema. B
* Registro de perfiles de usuario o administrador. B

# Features not to be Tested

This is a listing of what is NOT to be tested from both the Users viewpoint of what the system does and a configuration management/version control view. This is not a technical description of the software, but a USERS view of the functions.

Identify WHY the feature is not to be tested, there can be any number of reasons.

* Not to be included in this release of the Software.
* Low risk, has been used before and is considered stable.
* Will be released but not tested or documented as a functional part of the release of this version of the software.

Sections 6 and 7 are directly related to Sections 5 and 17. What will and will not be tested are directly affected by the levels of acceptable risk within the project, and what does not get tested affects the level of risk of the project.

# Approach (Strategy)

This is your overall test strategy for this test plan; it should be appropriate to the level of the plan (master, acceptance, etc.) and should be in agreement with all higher and lower levels of plans. Overall rules and processes should be identified.

* Are any special tools to be used and what are they?
* Will the tool require special training?
* What metrics will be collected?
* Which level is each metric to be collected at?
* How is Configuration Management to be handled?
* How many different configurations will be tested?
* Hardware
* Software
* Combinations of HW, SW and other vendor packages
* What levels of regression testing will be done and how much at each test level?
* Will regression testing be based on severity of defects detected?
* How will elements in the requirements and design that do not make sense or are untestable be processed?

If this is a master test plan the overall project testing approach and coverage requirements must also be identified.

Specify if there are special requirements for the testing.

* Only the full component will be tested.
* A specified segment of grouping of features/components must be tested together.

Other information that may be useful in setting the approach are:

* MTBF, Mean Time Between Failures - if this is a valid measurement for the test involved and if the data is available.
* SRE, Software Reliability Engineering - if this methodology is in use and if the information is available.

How will meetings and other organizational processes be handled?

# Item Pass/Fail Criteria

What are the Completion criteria for this plan? This is a critical aspect of any test plan and should be appropriate to the level of the plan.

* At the Unit test level this could be items such as:
  + All test cases completed.
  + A specified percentage of cases completed with a percentage containing some number of minor defects.
  + Code coverage tool indicates all code covered.
* At the Master test plan level this could be items such as:
  + All lower level plans completed.
  + A specified number of plans completed without errors and a percentage with minor defects.

This could be an individual test case level criterion or a unit level plan or it can be general functional requirements for higher level plans.

What is the number and severity of defects located?

* Is it possible to compare this to the total number of defects? This may be impossible, as some defects are never detected. o A defect is something that may cause a failure, and may be acceptable to leave in the application.
  + A failure is the result of a defect as seen by the User, the system crashes, etc.

# Suspension Criteria and Resumption Requirements

Know when to pause in a series of tests.

 If the number or type of defects reaches a point where the follow on testing has no value, it makes no sense to continue the test; you are just wasting resources.

Specify what constitutes stoppage for a test or series of tests and what is the acceptable level of defects that will allow the testing to proceed past the defects.

Testing after a truly fatal error will generate conditions that may be identified as defects but are in fact ghost errors caused by the earlier defects that were ignored.

# Test Deliverables

What is to be delivered as part of this plan?

* Test plan document.
* Test cases.
* Test design specifications.
* Tools and their outputs.
* Simulators.
* Static and dynamic generators.
* Error logs and execution logs.
* Problem reports and corrective actions.

One thing that is not a test deliverable is the software itself that is listed under test items and is delivered by development.

# Remaining Test Tasks

If this is a multi-phase process or if the application is to be released in increments there may be parts of the application that this plan does not address. These areas need to be identified to avoid any confusion should defects be reported back on those future functions. This will also allow the users and testers to avoid incomplete functions and prevent waste of resources chasing non-defects.

If the project is being developed as a multi-party process, this plan may only cover a portion of the total functions/features. This status needs to be identified so that those other areas have plans developed for them and to avoid wasting resources tracking defects that do not relate to this plan.

When a third party is developing the software, this section may contain descriptions of those test tasks belonging to both the internal groups and the external groups.

# Environmental Needs

Are there any special requirements for this test plan, such as:

* Special hardware such as simulators, static generators etc.
* How will test data be provided. Are there special collection requirements or specific ranges of data that must be provided?
* How much testing will be done on each component of a multi-part feature?
* Special power requirements.
* Specific versions of other supporting software.  Restricted use of the system during testing.

# Staffing and Training needs

Training on the application/system.

Training for any test tools to be used.

Section 4 and Section 15 also affect this section. What is to be tested and who is responsible for the testing and training.

# Responsibilities

Who is in charge?

This issue includes all areas of the plan. Here are some examples:

* Setting risks.
* Selecting features to be tested and not tested.
* Setting overall strategy for this level of plan.
* Ensuring all required elements are in place for testing.
* Providing for resolution of scheduling conflicts, especially, if testing is done on the production system.
* Who provides the required training?
* Who makes the critical go/no go decisions for items not covered in the test plans?

# Schedule

Should be based on realistic and validated estimates. If the estimates for the development of the application are inaccurate, the entire project plan will slip and the testing is part of the overall project plan.

* As we all know, the first area of a project plan to get cut when it comes to crunch time at the end of a project is the testing. It usually comes down to the decision, ‘Let’s put something out even if it does not really work all that well’. And, as we all know, this is usually the worst possible decision.

How slippage in the schedule will to be handled should also be addressed here.

* If the users know in advance that a slippage in the development will cause a slippage in the test and the overall delivery of the system, they just may be a little more tolerant, if they know it’s in their interest to get a better tested application.
* By spelling out the effects here you have a chance to discuss them in advance of their actual occurrence. You may even get the users to agree to a few defects in advance, if the schedule slips.

At this point, all relevant milestones should be identified with their relationship to the development process identified. This will also help in identifying and tracking potential slippage in the schedule caused by the test process.

It is always best to tie all test dates directly to their related development activity dates. This prevents the test team from being perceived as the cause of a delay. For example, if system testing is to begin after delivery of the final build, then system testing begins the day after delivery. If the delivery is late, system testing starts from the day of delivery, not on a specific date. This is called dependent or relative dating.

# Planning Risks and Contingencies

What are the overall risks to the project with an emphasis on the testing process?

* Lack of personnel resources when testing is to begin.
* Lack of availability of required hardware, software, data or tools.
* Late delivery of the software, hardware or tools.
* Delays in training on the application and/or tools.  Changes to the original requirements or designs.

Specify what will be done for various events, for example:

Requirements definition will be complete by January 1, 19XX, and, if the requirements change after that date, the following actions will be taken:

* The test schedule and development schedule will move out an appropriate number of days. This rarely occurs, as most projects tend to have fixed delivery dates.
* The number of test performed will be reduced.
* The number of acceptable defects will be increased.

o These two items could lower the overall quality of the delivered product.

* Resources will be added to the test team.
* The test team will work overtime (this could affect team morale).
* The scope of the plan may be changed.
* There may be some optimization of resources. This should be avoided, if possible, for obvious reasons.
* You could just QUIT. A rather extreme option to say the least.

Management is usually reluctant to accept scenarios such as the one above even though they have seen it happen in the past.

The important thing to remember is that, if you do nothing at all, the usual result is that testing is cut back or omitted completely, neither of which should be an acceptable option.

# Approvals

Who can approve the process as complete and allow the project to proceed to the next level (depending on the level of the plan)?

At the master test plan level, this may be all involved parties.

When determining the approval process, keep in mind who the audience is:

* The audience for a unit test level plan is different than that of an integration, system or master level plan.
* The levels and type of knowledge at the various levels will be different as well.
* Programmers are very technical but may not have a clear understanding of the overall business process driving the project.
* Users may have varying levels of business acumen and very little technical skills.
* Always be wary of users who claim high levels of technical skills and programmers that claim to fully understand the business process. These types of individuals can cause more harm than good if they do not have the skills they believe they possess.

# Glossary

Used to define terms and acronyms used in the document, and testing in general, to eliminate confusion and promote consistent communications.