**PROCESS**

I’ve been reading and getting familiar with all of the information about LLS, trying to understand the existing system to have a solid understanding of the functionalities and new features. And how all the components tigh together.

Now in order to implement a Quality Assurance process, we need to have a clear understanding of the following:

* The Scope and Objectives of the Project or Projects to developed. Defined Name and Goals.
* Any current Software development methodology and process follow by the company: Which agile methodology used and process framework.
* Teams and Roles: Roles already predefined and resources allocation.
* Initial Requirements: Business requirements and Functional requirements. (User Stories).
* Testing Methodology/Techniques. Type of testing to be performed:
* Functional testing
* Integration testing
* Regression testing
* System testing
* End to End testing.
* What are the problems, deficiencies faced in the current system, ej: existing portal is confusing for the customers.
* Any constrains (rules and limitations ej: time, resources) and Assumptions.
* Specific Success criteria (goals that satisfy the project, Process quality and product quality)
* Bug resolution and Bug Life Cycle
* Tools: Any tools used or planning to use:
* Playback tools
* Software for documentation
* Test Management tools
* Functional testing tools
* Performance testing tool
* Defect Tracker tool
* Project Time Line/Expected date for design completion.
* Standards and Process: changes process
* Risk items (risk assessment)

What stage we are on at this moment?

The QA gather all information necessary such as:

* Project Time Line/Expected date for design completion.
* A demo session, a walkthrough of the prototype to help us familiarize with the application
* Business requirements and Functional requirements.
* Architecture Documents
* Sample Test documentation
* Standards and Process: changes process
* Risk items (risk assessment)

1. If just Unit Testing is done, then start working on End to End testing to ensure that functionality is properly working.
2. I would start documenting in order to make easier to train new QA or end users.
3. Priorities
4. A demo session – a walkthrough of the prototype to help us familiarized with the application, a user for testing, to navigate through the site. I think you need to enter a valid info to navigate

I would produce a Test Strategy plan and would go from there, but would have to know of any ….. process used.

What is my envolvment as QA involvement from the beginning of the procect:

* Attend sprint-planning sessions
* Attend daily stand ups

Help to define……

Customized process of the Ecommer

* Access to test environment
* Swapping pixels
* Access to the TFS and not sure if the moved already to VSTS
* Testing environment credentials

1. What are the sources Leads come from? Email, social media, blogs etc. Online and offline sources (tv, Street marketing campaigns)
2. Buying leads? Are there rules for unwanted messages o.
3. Lead generation tool?
4. How manye Partners you have that provide leads?

Lead Gen: Encontrar potential customers

The lead us the same of vary depending on the partner?

* A visitor discover your business through one of your marketing channels
* CTA Call to Action: is an image button or message that calls website visitors to take some sort of action.
* Landing Page: is a web page a visitor lands or for a distinct purpose.
* Forms
* Offer
* Pixels drop cookies into customers or visiters browsers

To collect all relevant information in order to get a better understanding of the project and

From there, the new system requirements are defined. In particular, the deficiencies in the existing system are addressed with specific proposal for improvements.

The fists thing a QA needs to do is Understand the Product. Evaluate the existing system and its deficiencies identified, usually through interviewing system users and support personnel.

What is the scope of the project based in the information you sent me and the 6 points info

Now in order to implement a Quality Assurance process, we need to ensure that the whole team have a solid understanding of the functionality of new features before starting to work. IF YOU ALREADY HAVE IN PLACE………

Hi Huyen,

I want to let you know that I would be out of the office the 8th and 9th of September because I am taking a family vacation over those days. These was discussed and approved by GAP before accepting the job offer as I already have these holidays booked.

* The fists thing a QA needs to do is Understand the Product. Evaluate the existing system and its deficiencies identified, usually through interviewing system users and support personnel.

From there, the new system requirements are defined. In particular, the deficiencies in the existing system are addressed with specific proposal for improvements.

The QA gather all information necessary such as:

* Project Time Line/Expected date for design completion.
* A demo session, a walkthrough of the prototype to help us familiarize with the application
* Business requirements and Functional requirements.
* Architecture Documents
* Sample Test documentation
* Standards and Process: changes process
* Risk items (risk assessment)
* The QA team, at this stage the Test Planning starts. QA team review the documents and prepare the Test Strategy. Defined the Testing process and how it can fit into the client needs and the goal of the organization.

We must think always from a customer point of view.

* Creation of checklists
* Determine test environment requirements: hardware, software, communications.
* Determine testware requirements: Automation tools, Coverage analysis tools, test tracking and bug tracking etc.
* Input Data
* Identify tasks, roles
* Set schedule estimates
* smart-phone app.
* The different Modules of the software are identified. The Paths connecting all modules are identified. Testers write the Test Cases and finalizes test plans.

Test Cases are prepared based on the following scenarios:

* Positive scenarios
* Negative scenarios
* Boundary conditions
* Real World scenarios

Prepare test environment and testware, obtain user manuals/references, documents/configuration guides/ installation guides, set up test tracking process, set up test input data.

**Prepare following documents:**

* Test Strategy: After gathering the Requirements and identifying the scope of testing, will then put out a high level document called Test Strategy, defining the testing approach to achieve the test objectives.
* Scope and Objective
* Approach
* Roles and Responsibilities
* Tools identified for testing
* Metrics
* Risks and Mitigation plans
* Reporting
* Test Plan: document and have needed reviews/approvals.
* What, When, How and Who
* Timeframe
* Items to test and not to test
* Risks, issues, mitigation strategies and contingencies plans
* Define the methods and tools
* Pass and fail criteria
* Entry and Exit criteria
* Test Deliverables such as Test Plans, Test Cases and Test tools
* Environment setup
* Training
* Acceptance Criteria
* The different Modules of the software are identified. The Paths connecting all modules are identified. Testers write the Test Cases and finalizes test plans.

Test Cases are prepared based on the following scenarios:

* Positive scenarios
* Negative scenarios
* Boundary conditions
* Real World scenarios

Prepare test environment and testware, obtain user manuals/references, documents/configuration guides/ installation guides, set up test tracking process, set up test input data.

Once all documentation is delivered to the client to be approved. Once development is complete, that is, once the Build is made and deployed to the different testing environments, the QA team will begin the testing,

* Test Execution is done: Once Unit testing is completed and the code is released to QA, the functional testing is done.

bug reporting

Manual testing

Automation testing

Re-testing and Regression testing

* Final testing and deployment is done here and final report prepared: Once bugs are fixed, another release is given to QA with the modified changes. Regression testing is executed. Once the QA assures the software, this is released to Production. Testing will be done by the End Users

**How To implement QA Test Process for a large application**

* What is the current process
* What are the problems, deficiencies in the system that you are facing right now
* If just Unit Testing is done, then start working on End to End testing to ensure that functionality is properly working.
* I would start documenting in order to make easier to train new QA or end users.
* Think about optimization, automate much of the regression testing.

TC example:

Test Scenario: Validate the Admin login functionality

3 Test Cases:

* Login successful
* Login unsuccessful: when incorrect username is entered
* Login unsuccessful: when incorrect password is entered.

Each of the above test cases would have steps to address how we can check if a particular test condition is satisfied or not. EX, typing number, characters in the fields.

**How to prioritize software testing on little time**

**Have a prioritize list**

* What functionality is most important: what things will users do most often with this application
* What to the customer
* Which aspects can be tested earlier in the development cycle
* Highest risk aspects
* Which areas are most likely to contain show-stopping defects
* Any parts of the application governed by legal or regulatory agencies

Count the number of times a scenario appears in any of your categories. The more times the scenario appears, the higher the priority.

Scan the test cases. Note which ones are covered and which ones aren’t. on the ones that aren’t covered, ask yourself. Can I live with not testing this? If the answer is no, add it to the bottom of the list.

Do the testing: if you complete these tests before the time is up, do the same exercise again without repeating any scenario.

**Testing Tools**

That depends on the type of application and the type of testing to perform. If you need to test performance, you can not use a palyback tool, u need a performance testing tool such as load runner.

* Software for Documentation: Confluence, where you gather all related documentation which is used throughout the whole project. Any team member can see any changes as soon as any requirements or documents are changed, added, updated or delete.
* Test Management Tool: HP Quality Center
* Functional testing tool: Selenium, QTP
* Load Testing tool: LoadRunner
* Defect Tracker tool: Bugzilla, Jira

**Types of Testing:**

* Functional testing
* Integration testing
* Regression testing
* System testing
* End to End testing

**When to Automate**: Automation is good when the project is large, when there are many system users or when filling out a lot of forms. But the initial setup of automated test cases may take a while. Test automation tools can be expensive and are usually employed in combination with manual testing.

* Time and Budget.
* Large and critical projects
* When the same areas are required to be tested frequently
* Requirements not changing frequently
* Accessing the application for Load and Performance with many virtual users

*Planning Test Automation*

* How much do you want to automate? API? Front end? Full end to end?
* How is going to compliment other testing areas like unit tests, manual exploratory testing?
* Should focus on the most repetitive tasks and give testers more time to design tests/exploratory testing.
* How are you going to manage test data

***Other points to consider for setting up a QA function from scratch:***

* *Cross-browser, Cross device testing*
* *Mobile and Tablet Testing*
* *Parallel execution of automated tests*
* ***Exploratory Testing***
* *Tools, such as Jira, Jenkins, Selenium, etc…*
* *Continuous Improvement*
* *Recruitment of Testers*

**LIFELINE PROJECT**

Business Background: Screening tests, can find diseases and conditions early when they are easier to treat. Usually comprises of a combination of tests, which aim to give an individual a general overview of their health status and help to detect disease or risk factors early. Any issues can then be followed up with a GP, which is important as many people may suspect that they have a health issue but have not undergone medical tests.

The World Health Organisation (WHO) identifies eight key risks factors to health, all of which could be addressed in a health check:

* Smoking.
* Drinking alcohol.
* Obesity.
* High blood pressure.
* High cholesterol.
* High blood glucose.
* Poor diet.
* Lack of exercise.

**CSTAR: MMA**

The Maintenance management application (MMA) is a platform developed for the Customer Service Membership experts to review membership info and perform maintenance functions including:

* Display member info
* Display member plan info
* Display member benefits, history, status.
* Reinstate Membership
* Update billing info

Application Overview: Accessible through a link in our agent scheduling application CSTAR or directly through the web address: <http://yweb-na-app/LLSmembership/Member/>. Only users with specific security rights are granted access to the application.

**Ecom:** Consumer-oriented internet strategy

When you purchase a good or service online, you are participating in ecommerce. Help to stablish a market presence or enhance an existing market position by providing a cheaper and more efficient distribution chain for their products or services.

You can check out the various health screening package that are currently offered from the website. Once you choose the package and purchase them, you can make a payment online. A voucher for the health screening package that you have purchased will be emailed to you. The voucher is transferable, so that means you can even purchase the package for your parents or loved one and send them the voucher. Bring the voucher along when you go over to the hospital and claim the health screening without paying any extra.

* Build an Ecom that drives the customer quickly through the ecom process
* Incorporate clean design that is intentionally less distracting and more professional.
* Drive more conversions by simplifying the ecom process.
* Allow us to capture lost leads in an easy manner.
* Display member info
* Display member plan info
* Display member benefits, history, status.
* Reinstate Membership
* Update billing info

Application Overview: Accessible through a link in our agent

Sitecore: is a global software company offering two major products: a powerful CMS (Content Management System) and a fully-adaptive Digital Marketing System (DMS)

A content management system built on ASP.NET (**CMS**) is a software application or set of related programs that are used to create and manage digital content. The CMS incorporates a powerful desktop interface that is controlled by a fully-customizable role-based system.

Un CMS permite crear, editar y administrar los contenidos de nuestras paginas web. Con un Gestor de Contenidos un usuario puede crear, editar y actualizar todas las paginas web de su sitio, puede crear paginas nuevas, escirbir artículos en el blog, crear un portafolio de trabajos, montar un videoblog or un photolog.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. Asi es posible manejar el contenido y darle en cualquier momento un diseno distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores.

Trabajar con un Gestor de Contenido es fácil, normalmente detrás de un CMS hay una base de datos donde se organizan todos los contenidos de un sitio web. El usuario tiene acceso via navegador web a su panel de control online desde donde puede desde cualquier lugar y a tiempo real, modificar, crear y actualizar sus contenidos web.

Technology can be a catalyst to completely transform care at the delivery side by making it far more streamlined, prevent errors, avoid duplicates and make access to information better.

The fascinating part is if you accomplish that, then the patient can actually become a consumer of care

1. The application is responsive (web – app that is mobile-friendly) or native (mobile apps)?

Responsive: Requires internet connection

Poor performance

Lack of natural navigation

Lack of push notifications and other functionalities

Native: Requires internet connection not always necessary

Better performance

Increased Functionality

Enhanced security for users

Appcelerator Titanium – Is an open source mobile apps development framework by Appcelerator and licensed under Apache. This framework is coded to give developers an integrated environment to create an extensive range of mobile apps. It is based on JavaScript and entails HTML5, CSS3 as well as jQuery. Extensible development environment for creating beautiful native apps across different mobile devices and OS including iOS, Android and BlackBerry, as well as hybrid and HTML5.

*Story WorkShops: One of the most important activities in an agile project is frequent story workshop sessions. This is when the product owner, developers and testers gather in a room and start elaborating and fleshing out the details of the stories. This is important because everyone should have the same understanding of the story before starting the development work. Quality Assurance is about defect prevention rather than detection and so in the story workshops, the team get the chance to ask questions about the details of the story, any technical or design constraints and any blockers to developing the stories.*

*Here is a great opportunity to start writing out the acceptance criteria for the stories. Everyone should contribute and start thinking about the possible scenarios for each story, as each one will have a different idea, so the more heads on the story, the more scenarios can be thought of and the higher chance of preventing defects getting live.*

*Once everyone is certain on the detail and scope of each story, development starts.*

*When required, we should also perform non-functional testing, such as performance, load and security testing. Quite often the focus is on ensuring the functionality works well, however non-functional testing should be given the same priority, especially for web applications as they could be subjected to heavy load and / or attacks.*

*By performing non-functional testing, we can be sure that our application can handle load during peak times and that is not open to security threats.*

*ensure every user story is testable and includes* ***acceptance criteria****.*

*Build meaningful end-to-end test scenarios by utilizing trends, data and analytics from the production website to gather information about user activities and user journeys through the application.*

1. Google Analytics – tracking pixels

**Health Care Digital Future – to build a fully integrated healthcare IT system, always focusing on patience needs**

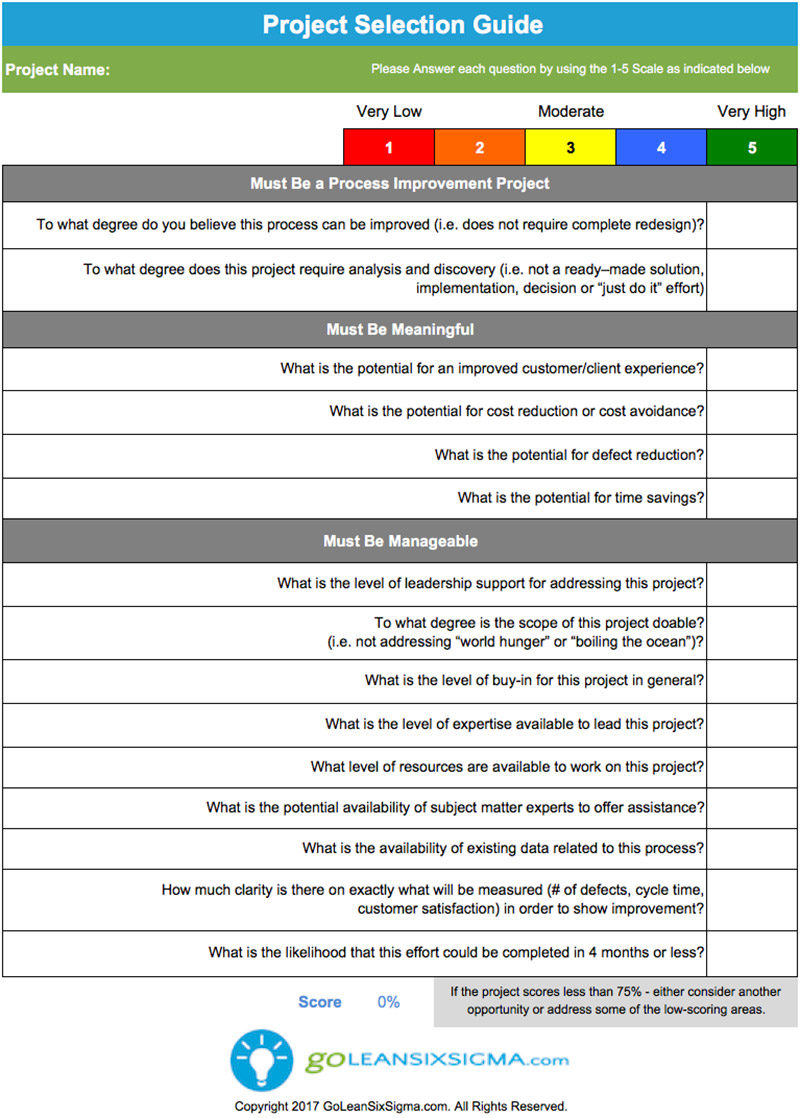
The idea is to automate highly standardized and repetitive tasks and to process vast amounts of statistical data.

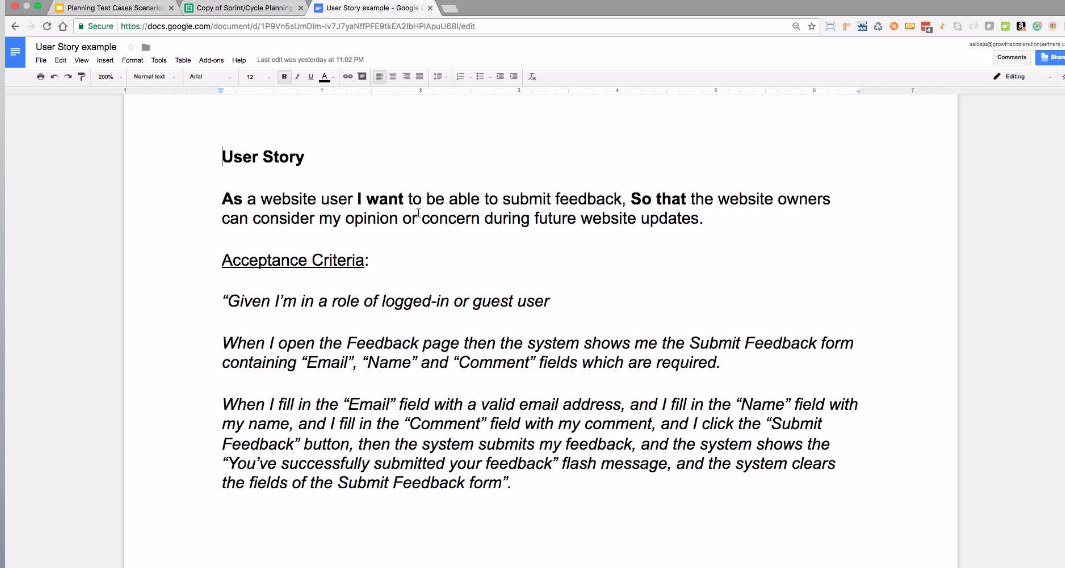
Patients around the world have grown more comfortable using digital networks and services. First of all, we need to understand what customers really want and from there, Products and Services need to be built based on that information.

Understand the patients’ digital preferences

* Services need to meet the patients needs
* Good quality
* Older patients prefer traditional digital channels such as websites and e-mails
* Efficiency and Better access to information
* Availability of a real person if the digital service doesn’t give them what they need.

he warehouse clubs are also looking at potential international expansion in markets such as Australia, Brazil, China, India, Mexico, Pakistan, South Africa and Turkey; th





Scenarios:

1. Email, name and comments fields are required
2. Email valid format
3. Submit button required
4. Confirmation message displays on success operation
5. Clear form after submitting data
6. Entered invalid data (invalid characters)

Test Cases: **Type of TC** – Sprint TC or Regression

**Priority:**

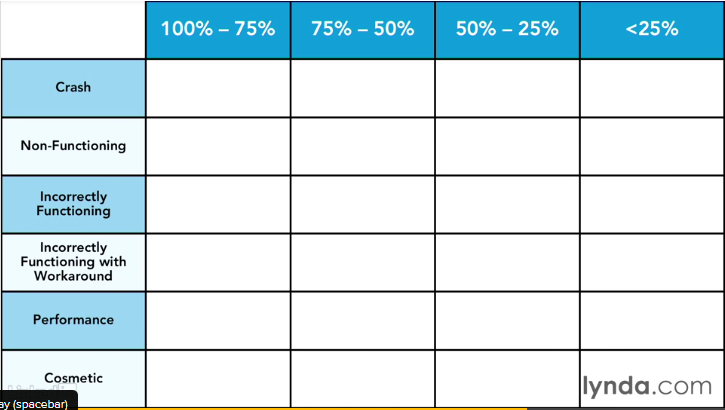
1. High Priority
2. High
3. High
4. Medium

**Execution Priority:**

1. High
2. High
3. High
4. Low

**Creation Time:**

1. 0,05 min



PRIORITY TABLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 100% - 75% | 75% - 50% | 50% - 25% | <25% |
| Crash | Priority 1 |  |  |  |
| Non-Functioning |  |  |  |  |
| Incorrectly Functioning |  |  |  |  |
| Incorrectly Functioning with Workaround |  | Priority 2 |  |  |
| Performance |  |  | Priority 3 or 4 |  |
| Cosmetic |  |  |  |  |

