Inventory Management

A NEW SOLUTION TO IQ TECHNOLOGY SOLUTIONS INVENTORY

Introduction

Our Purpose

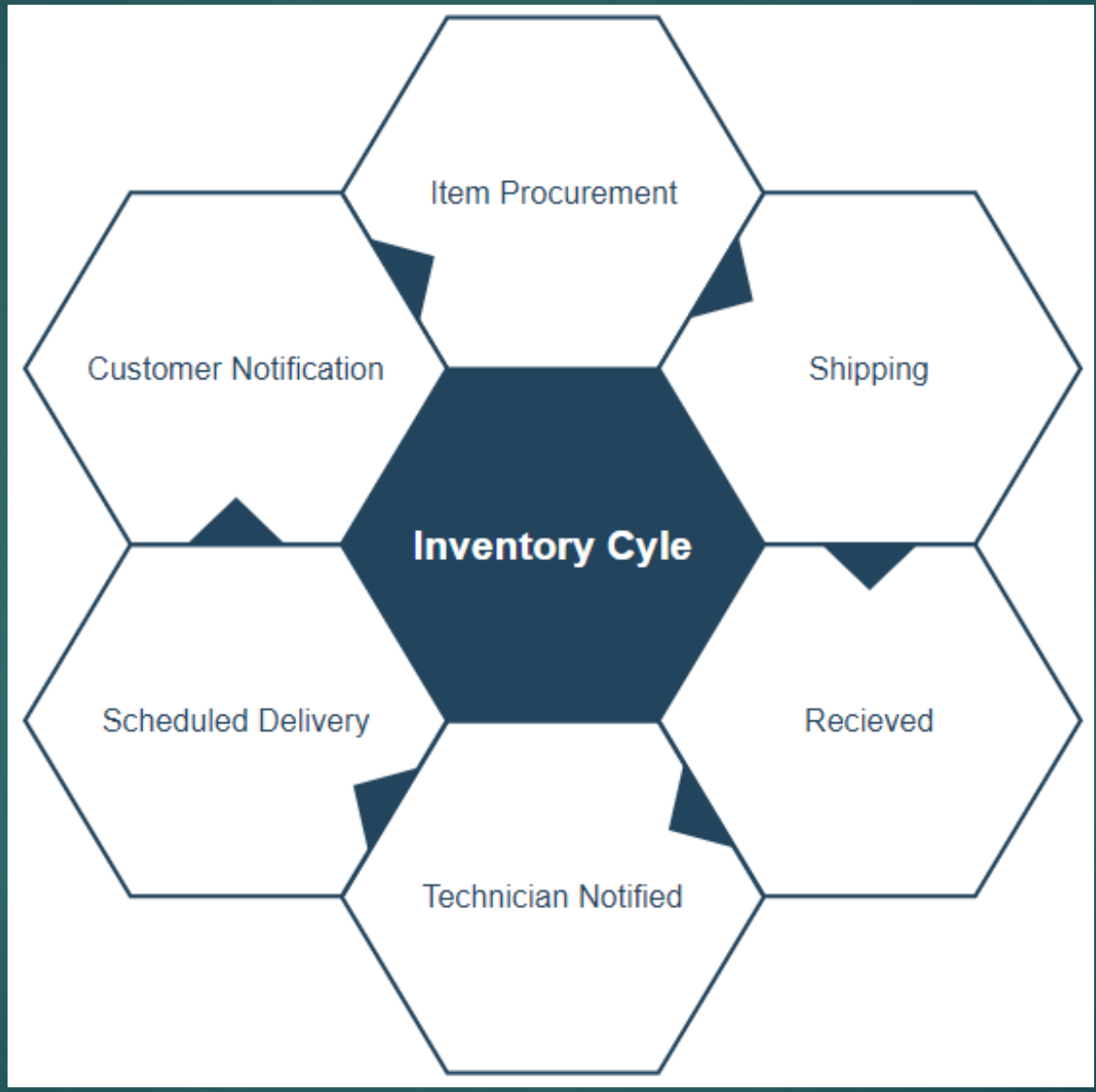
To insure improved employee efficiency and secure reliable client satisfaction

Scope

This project will affect all employees that access and handle inventory as well as clients waiting for their items

Overview

With a lack of inventory management and tracking, time is being wasted locating packages that should be readily accessible. A self-sustaining inventory system would solve this problem and reduce the time to identify and acknowledge received items.


Item Procurement
Customer Notification
Shipping
Inventory Cyle
Scheduled Delivery
Recieved
Technician Notified

Methods

- The Three Options Examined are:
  - Continue with the current inventory system
  - Purchase a pre built inventory management system
  - Build an inventory Management solution in house

Method Breakdown

1. Gathering and estimating data of the existing solution
2. Research needs to be done online to determine the cost of an existing system
3. Research for a DIY Solution

Results

OPTION 1: CONTINUE WITHOUT AN INVENTORY MANAGEMENT SYSTEM

Option 1 Results

- Pros
  - No Cost
- Cons
  - Items are often lost
  - Items are not tracked effectively and could be misplaced
  - Finding packages/items for clients is a hassle and is taking more time then necessary
    - This adds up to have a somewhat hidden cost for the company

Option 1 Results - Pricing

- Pricing would need to be calculated by monitoring employee time in regard to inventory tracking
- Hypothetically let's say there are 10 employees spending 1 hour per week tracking inventory with the current setup
- Let's assume a median pay of $15 per hour
- Adds up to $150 per week
  - $7,800 per year

Results

OPTION 2: PURCHASE AN EXISTING INVENTORY MANAGEMENT SYSTEM

Option 2 Results

- Pros
  - Can implement immediately
  - Comes with outside help/support if there is something wrong with the system
- Cons
  - Isn't purpose-built precisely for what would be needed
  - Can be high in cost
  - Any feature request will be vetted by the company that produces the product and may or may not end up in the final package

Option 2 Results – Pricing

- Oracle NetSuite
  - $99 for one user +$49 per additional user, per month
    - ~$1,200 per year, + ~$600 per additional user
- TradeGecko
  - $459 per month (8 users)
    - $5,508/year
- Cin7
  - Starting at $299/Month (amount of users not stated)
    - $3,588/year

Results

OPTION 3: HOMEGROWN SOLUTION

Option 3 Results

- Pros
  - Would fit the need exactly (Purpose built)
  - Features can be added/adjusted as needed
  - Items would be tracked exactly as needed
  - Time spent would be cut exponentially from current solution
- Cons
  - No outside support if there are problems.
  - Can be high cost to pay a developer, if there isn't already one on the team
  - Cannot implement solution immediately, and the cost will most likely be front loaded (pay a lot up front for server hardware or cloud server space)

Option 3 Results – Hosting Pricing

- Option 1 (Self Hosted) ~ $2,750 ~ one time cost
  - Server Cost: $2773.55
  - This option would not be externally accessible, which keeps the hardware/internet cost down, but also doesn't allow for anyone without VPN access to the network to access the software. So, if a portal for clients would ever need to be built the server would have to be setup to be accessible from the internet which would drive up the costs (Cert, static IP from ISP, Beefed up network security, etc.)
- Option 2 (AWS/Cloud)
  - T2 medium from amazon, 1 year standard pricing
  - $270/year
  - This option could be easily made externally accessible, but would still need the SSL certificate, as well as a domain (~$250/year)

Option 3 – Development Stack

- LAMP
  - (pretty close to) Universally supported
  - Oldest stack (I think)
  - Runs on freeware (Linux, Apache, MySQL, PHP)
  - Guides exist to help get this started

Option 3 – Development Stack

- MEAN
  - Newer
  - More robust
  - Runs on freeware (MongoDB, Express.js, Angular.js, Node.js)
  - Not as many developers, so they will be more expensive
  - Nothing found is using this stack in the open source department, for inventory management.

Option 3 – Development Stack

- ASP.NET MVC 5 + React/Redux
  - Similar benefits to the MEAN stack
  - Server coded in C# (more common language)
  - Built for/Only runs on windows stack (Windows Server, IIS, MS-SQL, ASP.NET)
  - Nothing found is using this stack in the open source department, for inventory management.

Option 3 – Feature Set

- Add item types
- Add items of that type with a location
- Tie specific boxes/items to specific clients
- Update inventory
  - Possibly automatically w/ barcode scanners?
- Browser based, so compatibility with different operating system won't be an issue.
- Super user friendly, to save time
- User logins to track who is removing/adding items to inventory to help with audits.

Option 3 - Timeline

- Sprint 1
  - Setup site hosting
- Sprint 2
  - Launch site on stack with basic demo application
- Sprint 3
  - Implement item types, adding items of that type and updating inventory
- Sprint 4
  - Add users/logins, ability to tie items to clients
- Sprint 5
  - Clean up/build user interface
- Sprint 6
  - Build in help/instructions into screens

Option 3 - Developer options

- Utilize an existing developer
  - Would allow a developer to be familiar with the program for future enhancements
  - This person is already being paid by the company
  - Would take time from other projects
- Hire a freelance developer/outside developer company
  - This would likely cost $5,000-$10,000 for the initial app, but then a person inside the company could be trained on it for support purposes
- Hire an in house developer
  - This would make for the best support
  - Wouldn't take time away from other projects
  - Would be the most expensive.

Ranking and Recommendations

Ranking

1. DIY Solution (option 3)
2. Buying existing software (option 2)
3. Continue current method (option 1)

Recommendation – Hosting

- We recommend the AWS option for hosting, as it's more future-proof than a new business server and should cost less in the end.

Recommendation – Development Stack

- We recommend using the LAMP stack since there are existing open source solutions to this that can be used and modified to fit the exact business need. This will also reduce upfront time/cost.

Recommendation – Feature Set

- Add item types
- Add items of that type with a location
- Tie specific boxes/items to specific clients
- Update inventory
  - Possibly automatically w/ barcode scanners?
- Browser based, so compatibility with different operating system won't be an issue.
- Super user friendly, to save time
- User logins to track who is removing/adding items to inventory to help with audits.

Recommendation - Timeline

- Sprint 1
  - Setup site hosting
- Sprint 2
  - Launch site on stack with basic demo application
- Sprint 3
  - Implement item types, adding items of that type and updating inventory
- Sprint 4
  - Add users/logins, ability to tie items to clients
- Sprint 5
  - Clean up/build user interface
- Sprint 6
  - Build in help/instructions into screens

Recommendation – Developer

- We recommend the usage of a freelance developer in combination with a current employee. The heavy lifting can be outsourced, but the knowledge and understanding will remain with the existing team member.

Presentation Video

HTTPS://YOUTU.BE/XX1WSCNA99Y